Editora Poisson

CAPACITAÇÃO EM ROBÓTICA INDUSTRIAL

FERRAMENTAS ACADÊMICAS PARA SOLUÇÕES INDUSTRIAIS

2ª EDIÇÃO

Organizadores:
José Pinheiro de Queiroz Neto
José Fábio Lima Nascimento
Nelson Alexandre Gouvea
Dominique Vieira de Carvalho

LG inova POLO DE INOVAÇÃO IFAM FAEPI

INSTITUTO FEDERAL
Amazonas
Campus Manaus-Distrito Industrial

Organizadores
José Pinheiro de Queiroz Neto
José Fábio Lima Nascimento
Nelson Alexandre Gouvea
Dominique Vieira de Carvalho

Capacitação avançada em robótica industrial: ferramentas acadêmicas para soluções industriais

2ª Edição

Belo Horizonte
Editora Poisson
2026

Editor Chefe: Dr. Darly Fernando Andrade

Conselho Editorial

Dr. Antônio Artur de Souza – Universidade Federal de Minas Gerais
Ms. Davilson Eduardo Andrade
Dra. Elizângela de Jesus Oliveira – Universidade Federal do Amazonas
MSc. Fabiane dos Santos
Dr. José Eduardo Ferreira Lopes – Universidade Federal de Uberlândia
Dr. Otaviano Francisco Neves – Pontifícia Universidade Católica de Minas Gerais
Dr. Luiz Cláudio de Lima – Universidade FUMEC
Dr. Nelson Ferreira Filho – Faculdades Kennedy
Ms. Valdiney Alves de Oliveira – Universidade Federal de Uberlândia

Dados Internacionais de Catalogação na Publicação (CIP)

C236
Capacitação avançada em robótica industrial: ferramentas acadêmicas para soluções industriais / Organização: José Pinheiro de Queiroz Neto, José Fábio Lima Nascimento, Nelson Alexandre Gouvea, Dominique Vieira de Carvalho
Belo Horizonte– MG: Editora Poisson, 2026

Formato: PDF
ISBN: 978-65-5866-604-2
DOI: 10.36229/978-65-5866-604-2
Modo de acesso: World Wide Web
Inclui bibliografia

1.Robótica 2.Educação I. NETO, José Pinheiro de Queiroz II. NASCIMENTO, José Fábio Lima III. GOUVEA, Nelson Alexandre IV. CARVALHO, Dominique Vieira de V.Título

CDD-370

Sônia Márcia Soares de Moura – CRB 6/1896

O conteúdo deste livro está licenciado sob a Licença de Atribuição Creative Commons 4.0.
Com ela é permitido compartilhar o livro, devendo ser dado o devido crédito, não podendo ser utilizado para fins comerciais e nem ser alterado.

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos seus respectivos autores.

Esse e outros títulos podem ser baixados gratuitamente em www.poisson.com.br
Entre em contato pelo contato@poisson.com.br

Organizadores

José Pinheiro de Queiroz Neto

Formado em Engenharia Elétrica e com Especialização em Engenharia de Produção pela Universidade Federal do Amazonas, desenvolveu os cursos de Mestrado e Doutorado em Ciência da Computação na Universidade Federal de Minas Gerais e Pós-doutorado em Educação Profissional na Tampere University of Applied Science (TAMK) / Finlândia.. Como profissional, atuou em empresas do Parque Industrial de Manaus, desenvolvendo atividades na área de Engenharia Industrial, tendo sido Chefe de Engenharia Industrial, na Gradiente Eletrônica S/A. Na área acadêmica, foi coordenador dos cursos Superiores do IFAM/Campus Distrito Industrial, Professor colaborador dos Mestrados em Ciência da Computação (UFAM) e Engenharia Elétrica (UFAM), Diretor Geral do Campus Distrito Industrial e Pró-Reitor de Pesquisa, Pós-Graduação e Inovação do do Instituto Federal de Educação, Ciência e Tecnológica do Amazonas (IFAM). Foi General Chair do XIX SIBGRAPI (Brazilian Symposium on Computer Graphics) em 2006. Participou como bolsista CNPq do Programa Professores para o Futuro - Finlândia. Atualmente é professor pesquisador do IFAM, professor do Mestrado/Doutorado PPGET, Membro da Associação Polo Digital de Manaus (APDM) e do COnselho Superior do IFAM. Na área de pesquisa, participa de grupos voltados a área de Ensino, com particular interesse em ferramentas modernas de ensino e educação, e grupo de pesquisa na área de Ciência da Computação/Automação, com particular interesse em Visão Computacional, Robótica e Sistemas Embarcados.

José Fábio de Lima Nascimento

Graduado em Engenharia Mecânica pela Universidade Federal da Paraíba (2001), Especialização em Dinâmica e Controle de Sistemas Robóticos pela Universidade Federal do Amazonas (2007), Mestrado em Engenharia Mecânica pela Universidade Federal da Paraíba (2004). Doutor em Engenharia de Recursos Naturais da Amazônia (ITEC/UFPA), professor do Ensino Básico, Técnico e Tecnológico (EBTT) do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas e Diretor Administrativo e Financeiro do Polo de Inovação de Manaus / IFAM. Tem experiência na área de engenharia mecânica, com ênfase em controle de sistemas mecânicos, atuando principalmente nos seguintes temas: Automação, robótica industrial, modelagem dinâmica de sistemas mecânicos, síntese e aplicação de semicondutores inorgânicos, principalmente, no desenvolvimento de estudos que envolvem as técnicas de caracterizações: Difração de raios X (DRX), refinamento Rietveld, espectroscopia vibracional Raman e no infravermelho, espectroscopia UV-vis de sólidos, microscopia eletrônica de varredura (MEV), termogravimetria (TGA) e calorimetria exploratória diferencial (DSC). Tem estudado ao longo dos últimos anos, as propriedades morfológicas, estruturais e ópticas de materiais obtidos pelo método de moagem de alta energia, investigando propriedades de interesse em aplicações no campo de fotocatálise, obtenção de biocombustíveis, armazenamento de energia (baterias de estado sólido), semicondutores e sensores.

Organizadores

Nelson Alexandre Gouvea

Graduado em Administração de Sistemas pelo Centro Universitário de Ensino Superior do Amazonas (2002). Atualmente é gerente geral da LG ELECTRONICS DO BRASIL(Manaus) . Tem experiência na área de Engenharia de Produto com ênfase em produtos de TV, Audio e Video e ar condicionado. Gerencia os projetos de P&D controlados pela SUFRAMA e também atua como representante da LG Manaus no relacionamento governamental em associações e instituições governamentais.

Dominique Vieira de Carvalho

Possui graduação em Engenharia de Produção pela Universidade do Estado do Amazonas (2024). Tem experiência na área de Engenharia de Produção, com ênfase em Gerência do Projeto e do Produto. Foi bolsista de projetos de pesquisa e desenvolvimento através da Lei de Informática de projetos da Universidade do Estado do Amazonas voltados para as áreas de capacitação em tecnologias digitais e engenharia de software. É atualmente Analista de Pesquisa e Desenvolvimento na LG Eletrônicos do Brasil e consultor na área de Gestão de Produtos.

Prefácio

A sociedade global vive, na atualidade, uma urgência tecnológica e digital, cabendo, pois, às instituições de ensino desenvolver ações que oportunizem uma integração entre a indústria e a academia.

Hoje, a automação e a inteligência das máquinas não são mais apenas tendências e sim realidades que moldam o presente e definem o futuro da indústria. Neste contexto, a robótica industrial ocupa um papel central, exigindo profissionais cada vez mais capacitados, versáteis e conectados com as inovações tecnológicas.

O Instituto Federal do Amazonas, ao longo dos anos, vem desenvolvendo pesquisas inovadoras cujo foco é o aprimoramento de tecnologias voltadas para o a melhoria dos serviços industriais. Entre os inúmeros projetos e parcerias, destacamos a LG Electronics do Brasil que vem cumprindo com maestria sua missão em promover com excelência a Educação, Ciência e Tecnologia para o desenvolvimento sustentável da Amazônia.

O desenvolvimento de diversos projetos de formação tecnológica à comunidade, oportunizados pelas pesquisas desenvolvidas por estudiosos de diversas áreas, contribui para a Formação Inicial e Continuada (FIC), por meio de parcerias com a indústria, o comércio, organizações sociais e outras esferas governamentais.

A obra **Capacitação avançada em robótica industrial: ferramentas acadêmicas para soluções industriais - 2ª edição**, é fruto de uma profunda dedicação à formação avançada em robótica industrial. O autor construiu um percurso didático que une teoria e prática, academia e chão de fábrica, inovação e aplicabilidade. Cada capítulo foi pensado para oferecer ao leitor não apenas conhecimento técnico, mas também ferramentas para pensar soluções criativas e eficientes diante dos desafios da automação.

Esta obra traz um aporte teórico-prático que vai das Múltiplas competências, utilizando o Design Thinking à Visão Computacional; da Manufatura Aditiva à Programação em CLP e IHM, aliado a conhecimentos aprofundados em Robótica Industrial. A linguagem e a metodologia utilizadas apresentam com clareza e profundidade os temas que compõem o universo da robótica moderna, leitura essencial para estudantes, engenheiros, técnicos e todos aqueles que desejam se posicionar na vanguarda da transformação industrial.

Caros leitores, mais do que um manual, esta obra é um convite à construção de um novo olhar sobre a robótica — um olhar que integra saberes, conecta disciplinas e prepara mentes para o futuro. Portanto, tenham uma boa leitura e aproveitem todos os conhecimentos e descobertas que esse material oportunizará aos senhores.

"O Conhecimento desvela, liberta, fortalece e oportuniza."

Dra. Maria Francisca Morais de Lima

Pró-Reitora de Extensão do IFAM

SUMÁRIO

Capítulo 1: Curso capacitação em robótica industrial: conexão entre a prática da indústria e os saberes da academia 08

José Cavalcante Lacerda Junior, Vitor Bremgartner da Frota

DOI: 10.36229/978-65-5866-604-2.CAP.01

Capítulo 2: Disciplina de múltiplas competências e *hands on* 20

Ana Cláudia Ribeiro de Souza, José Pinheiro de Queiroz Neto

DOI: 10.36229/978-65-5866-604-2.CAP.02

Capítulo 3: Manufatura aditiva 30

Renan Cavalcante Santos, Davi da Silva Ribeiro Castro

DOI: 10.36229/978-65-5866-604-2.CAP.03

Capítulo 4: Introdução á robotica industrial 41

Benjamin Batista de Oliveira Neto, Ricardo Augusto Medeiros de Oliveira

DOI: 10.36229/978-65-5866-604-2.CAP.04

Capítulo 5: Disciplina de programação para aplicações robóticas 60

Lenildo Marcos da Mota Silva, Ricardo Augusto Medeiros de Oliveira

DOI: 10.36229/978-65-5866-604-2.CAP.05

Capítulo 6: Programação em CLP 90

Hugo Alves Velozo, Vanderson de Lima Reis

DOI: 10.36229/978-65-5866-604-2.CAP.06

Capítulo 7: Programação em *Interface* Homem Máquina - IHM 110

Vanderson de Lima Reis, Hugo Alves Velozo

DOI: 10.36229/978-65-5866-604-2.CAP.07

Capítulo 8: Visão computacional e aplicações industriais 125

Marcelo Chamy Machado, Anderson Gadelha Fontoura

DOI: 10.36229/978-65-5866-604-2.CAP.08

Capítulo 9: Inovações tecnológicas para automação avançada 151

Diego Câmara Sales

DOI: 10.36229/978-65-5866-604-2.CAP.09

CAPÍTULO

01 Curso capacitação em robótica industrial: conexão entre a prática da indústria e os saberes da academia

José Cavalcante Lacerda Junior[1], Vitor Bremgartner da Frota[2]

Os processos formativos no âmbito da educação e a integração com o mundo do trabalho é uma realidade imperativa. A velocidade das informações, a automação dos artefatos, o emprego de manufaturas aditivas e visão computacional empreendem uma nova articulação entre o universo da academia e a realidade fabril. Um dos pontos fundamentais para essa articulação é a identificação de profissionais qualificados, o que favorece uma economia de tempo para aqueles que empregam, bem como favorece uma maior integração de tais profissionais em equipes (Akdur, 2022).

Outro aspecto interessante de tal articulação é que os saberes problematizados devem ser respaldados nos contextos que envolvem as distintas realidades do discente com vistas a tornar o aprendizado mais proveitoso. O processo de ensino e aprendizagem não pode estar desconectado desta conjuntura, pois, o conhecimento é um processo construído e voltado para a solução de problemas.

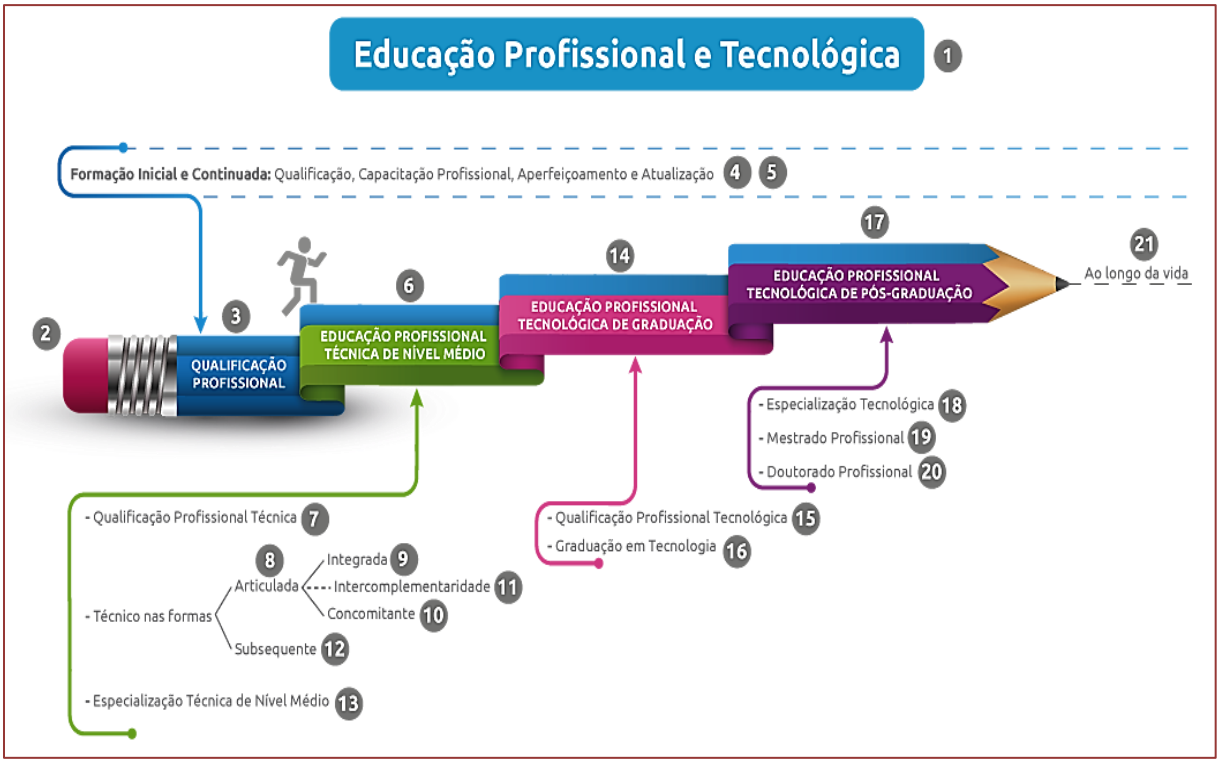
Essas conjunturas não devem escapar ao contexto da Educação Profissional e Tecnológica (EPT). Fundamentalmente, a EPT intenciona qualificar profissionais para os diferentes setores da economia, em colaboração com os setores produtivos e a sociedade, garantindo a efetivação e o fortalecimento dos arranjos produtivos locais, além de incentivar e apoiar processos educacionais que visem à criação de empregos e geração de renda.

No âmbito do Brasil, a EPT está inserida na Lei de Diretrizes e Bases – LDB (Brasil, 1994) e redimensionada por intermédio da Lei 11.741 (Brasil, 2008). Sua estruturação envolve três formas de curso: 1) Formação Inicial e Continuada ou Qualificação Profissional – FIC; 2) Educação Profissional Técnica de Nível Médio – EPTNM; 3) Educação Profissional Tecnológica de graduação; 4) Educação Profissional Tecnológica de pós-graduação, conforme se observa na Figura 1.1.

[1] Doutor em Ciências do Ambiente e Sustentabilidade na Amazônia. Apoio Pedagógico do Curso de Capacitação Avançada de Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM. E-mail: *jose.cavalcante@ifam.edu.br*

[2] Doutor em Informática. Orientador de alunos do Curso Capacitação Avançada em Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM. E-mail: *vitorbref@ifam.edu.br*

Figura 1.1 - Organização da EPT



Fonte: MEC (2018).

Os cursos desenvolvidos no contexto da Formação Inicial e Continuada – FIC são, de modo geral, organizados com o intuito de aprimorar habilidades para a vivência no mundo do trabalho e na sociedade. Tais cursos envolvem desde atualização, aperfeiçoamento até capacitação que fomentem a qualificação profissional. Sua estruturação ocorre por intermédio do estabelecimento de uma carga horária e definição de público de acordo com a finalidade da proposta.

É nesse contexto que surge o Curso de Capacitação Avançada em Robótica Industrial. Sua articulação é parceria entre o Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM) campus Manaus Distrito Industrial, Polo de Inovação e a empresa LG *Electronics*. Em síntese, o projeto coaduna esforços do setor público e privado com um objetivo comum: capacitar profissionais qualificados para os desafios do mundo do trabalho, especialmente no contexto da Indústria 4.0.

Mais do que apenas oferecer uma formação técnica, esse curso é uma ponte entre a demandas reais do setor industrial com os diversos saberes promovidos pela academia, neste caso no campo da Engenharia e Tecnologias, promovendo um diálogo essencial sobre temas como programação, manufatura aditiva, interface homem e máquina, visão computacional, etc.

1.1. ESTRUTURAÇÃO DO CURSO

A principal missão do curso é oferecer uma formação prática e atualizada em robótica industrial, com foco nas tecnologias emergentes e nas demandas específicas do Polo Industrial de Manaus. Para isso, a proposta articula diversas ações integradas, como:

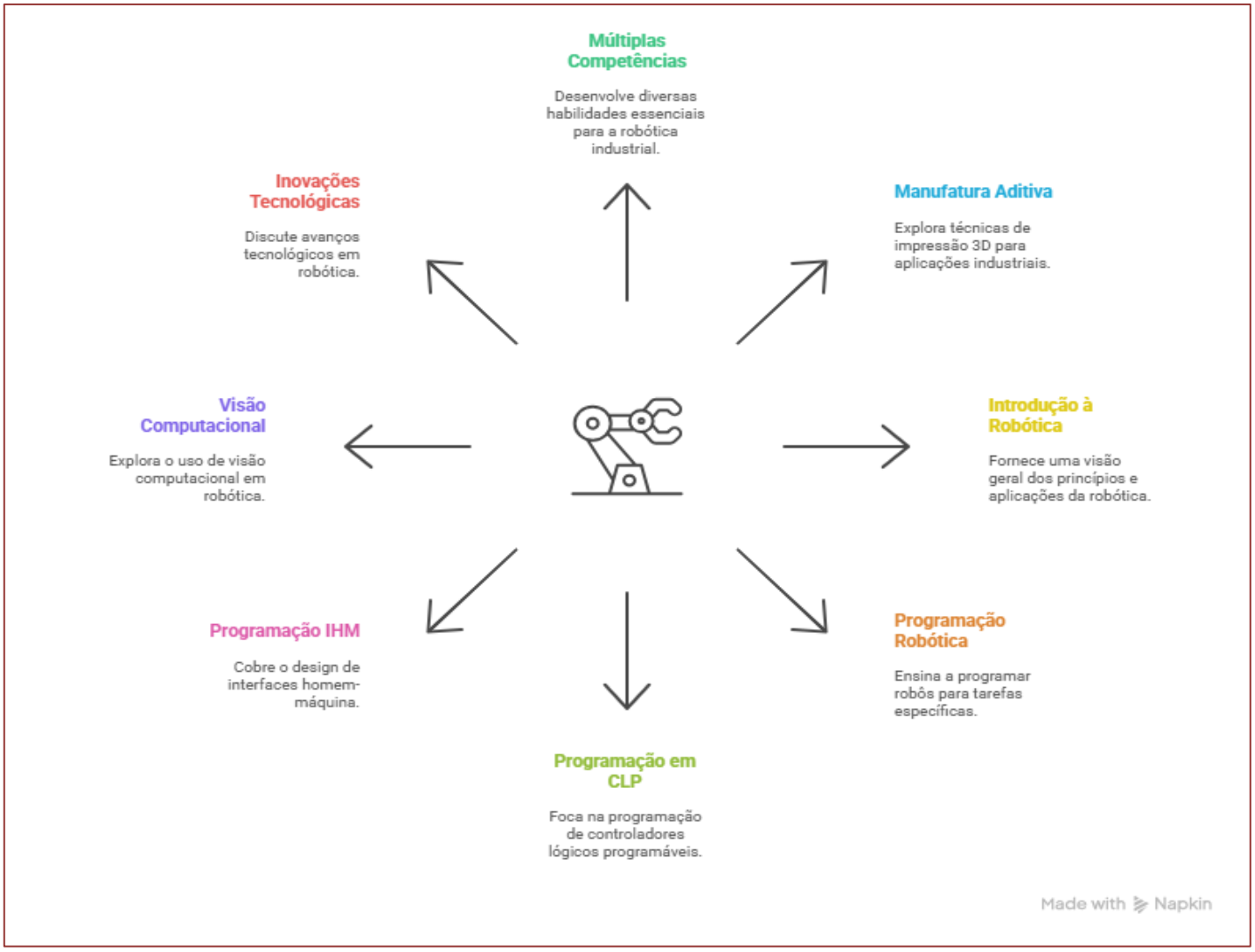
- aproximar o ambiente acadêmico da realidade industrial;
- capacitar os estudantes por meio da Aprendizagem Baseada em Projetos (PBL);
- desenvolver materiais didáticos alinhados às tecnologias utilizadas na indústria;
- preparar os docentes tecnicamente para o uso dessas tecnologias;
- estimular a criação de soluções práticas e inovadoras, como os MVPs (produtos mínimos viáveis).

A metodologia adotada, a PBL (Problem-Based Learning), coloca o estudante no centro do processo de aprendizagem. Nessa abordagem, os alunos são desafiados a resolver problemas reais, trabalhando em equipe e integrando diferentes saberes. O conhecimento, assim, deixa de ser apenas teórico e ganha forma e prática por intermédio de vivência dos desafios empreendidos na trilha de capacitação dos módulos e da trilha Jornada *hands on.*

Trilha 1 – Capacitação em Módulos

Essa etapa é composta por disciplinas teóricas e práticas, organizadas nos seguintes módulos: i) Múltiplas Competências e *Hands On (40h); Manufatura Aditiva (60h);* Introdução à Robótica Industrial (60h); Programação para Aplicações Robóticas (60h); Programação em CLP (60h); Programação IHM (60h); Visão Computacional e Aplicações Industriais (60h); Inovações Tecnológicas para Robótica Avançada (20h), conforme se ilustra na Figura 1.2.

Figura 1.2 - Trilha 1: Capacitação

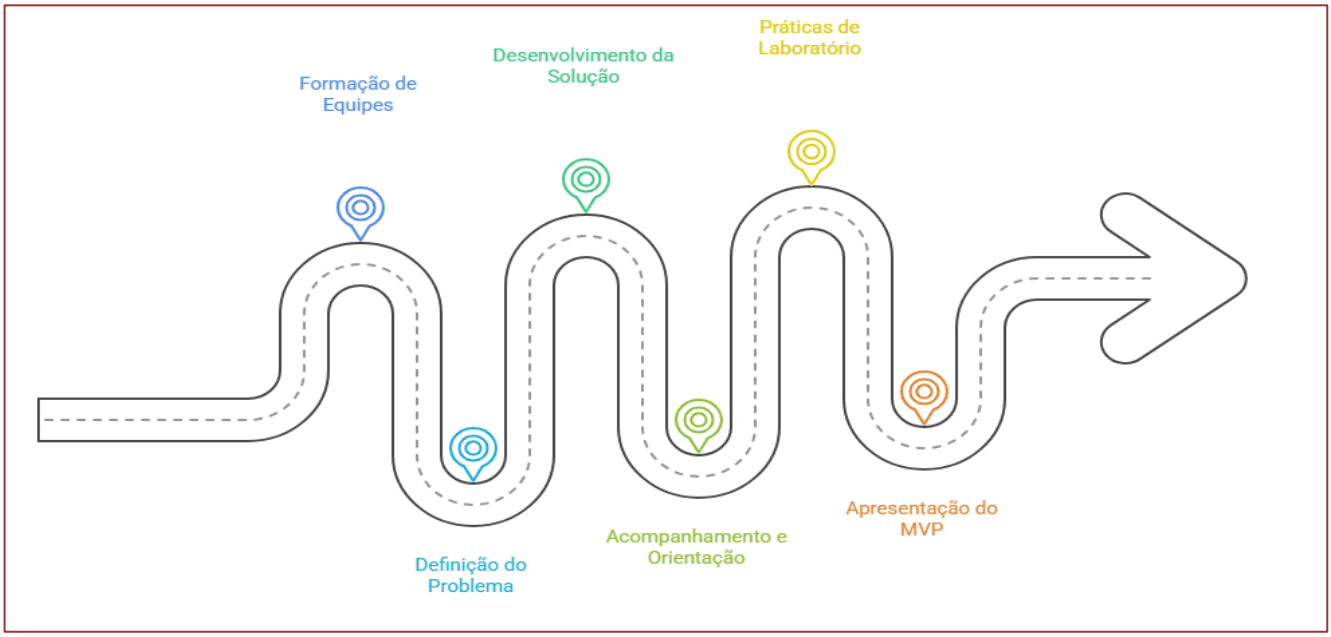


Fonte: Adaptado de Napkin (2025).

Trilha 2 – Jornada Hands On

Em síntese, essa trilha põe em prática o que foi aprendido. Os alunos são divididos em grupos e desafiados a resolver problemas reais da indústria, sugeridos pela empresa LG *Eletronics*. A jornada é composta por diversas etapas, desde a formação das equipes até a construção de um Produto Mínimo Viável – MVP, conforme se observa na Figura 1.3. Todo esse processo é acompanhado por professores e um orientador com encontros, reuniões e práticas de laboratório que culminam na apresentação de uma solução funcional.

Figura 1.3 - Trilha 2: Jornada



Fonte: Adaptado de Napkin (2025).

1.2. ONDE TUDO ACONTECEU?

As atividades do curso foram realizadas no INOVA – Polo de Inovação do IFAM, mais especificamente no Laboratório de Robótica Industrial (LRI), um espaço construído para a realização deste. O LRI foi inaugurado em 2023 e no mesmo ano desenvolveu a primeira versão deste curso. No ano de 2025, o LRI voltou a ofertar a capacitação com a ampliação do itinerário formativo a partir da inserção das disciplinas de Manufatura Aditiva e Inovações Tecnológicas.

Outro elemento importante nesta versão diz respeito aos participantes do curso. Na primeira versão, o público-alvo ficou dimensionado entre alunas da graduação e profissionais da área da engenharia e tecnologia. Nesta segunda versão, houve a abertura para inserção de profissionais técnicos que já atuam na indústria para participar do processo de capacitação. Esse incremento trouxe maior amplitude na troca das experiências entre os alunos, de forma específica, na construção de estratégias para viabilizar as resoluções dos desafios.

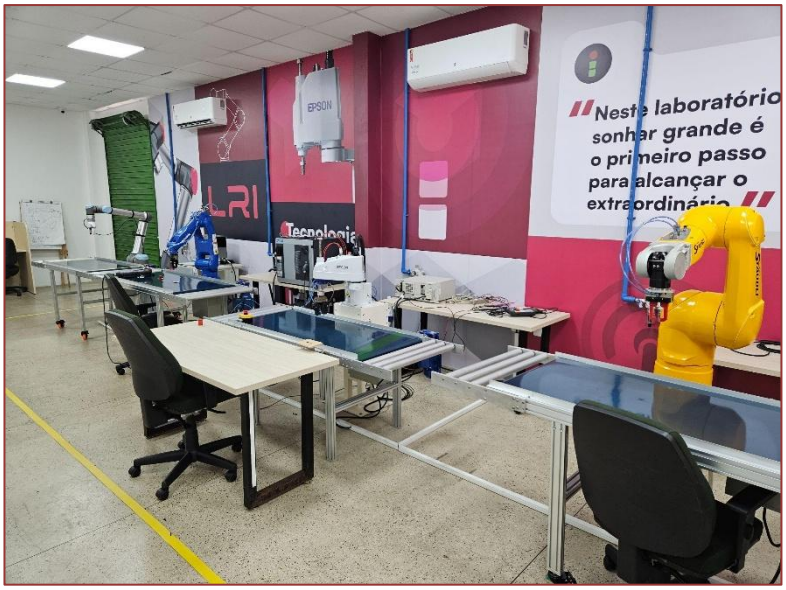
No que diz respeito à estrutura física, o LRI está organizado em dois espaços. No primeiro, conforme se observa na Figura 1.4, está destinado para o espaço de acolhimento dos alunos e realização das aulas expositivas, realização das pesquisas e encontros dos grupos formados para atuar nos desafios.

Figura 1.4 - Espaço 1

Fonte: Os autores, 2025.

No segundo espaço, conforme se observa a Figura 1.5, está destinado às capacitações práticas, onde ocorre o manuseio empírico dos quatro (04) braços robóticos (da direita para esquerda, Stäubli, EPSON, Yaskawa e Universal Robots) utilizados para a Capacitação. É nesse espaço, também, que os desafios são testados e validados pelos alunos no decorrer da jornada H*ands On*.

Figura 1.5 - Espaço 2



Fonte: Os autores, 2025.

Por fim, é importante ressaltar que a implementação do LRI no curso de capacitação representa um significativo avanço no processo de formação prática e técnica dos participantes. Ao constituir-se como *lócus* de aprendizagem ativa, o LRI foca em experimentação, inovação e solução de desafios experimentados pelo setor industrial. Para além dessa condição, os alunos participantes do curso podem desenvolver projetos integrados e melhorar habilidades fundamentais como programação, automação e gestão de sistemas robóticos, em sintonia com os elementos da Indústria 4.0.

1.3. PARCERIAS E TRANSFORMAÇÃO

A realização do curso é uma parceria entre a indústria e academia. A relação fluida e direta entre esses dois segmentos não somente mobiliza conhecimentos como expõe práticas e estratégias que fomentam um processo formativo atento às demandas do contexto contemporâneo. Não é à toa que vivemos um tempo em que o conhecimento técnico precisa estar em sintonia com as transformações do mercado e com os avanços tecnológicos.

Esse pressuposto fundamenta e orienta o curso de robótica. O processo formativo dos estudantes em EPT se consolida, ainda mais, quando o processo consegue interligar o que se aprende em sala de aula com a resolução de problemas reais. Essa dinâmica oportuniza, ainda, ao estudante a efetivação do postulado "aprender a aprender" (Gargallo López, 2020), o qual pode fomentar uma postura formativa e proativa diante dos desafios profissionais e da vida.

E ainda, a capacidade de aprender possibilita aos participantes a aquisição de independência intelectual e adaptabilidade às constantes inovações tecnológicas do campo. Aprender a aprender envolve a capacidade de procurar, escolher, entender e utilizar novos conhecimentos de maneira crítica e incessante, algo crucial em um campo caracterizado por mudanças rápidas e pela exigência de atualização contínua.

Nesse cenário, as parcerias entre instituições de ensino e empresas são cada vez mais necessárias. Segundo Martino (2006), é justamente essa conexão que ajuda a superar o antigo dilema entre teoria e prática, onde os atores envolvidos estabelecem pouco diálogo e o processo é entendido sob uma ótica tecnicista em ambos os cenários.

Parcerias como essa, com efeito, podem ajudar na formação de um ambiente de colaboração, onde ideias se cruzam, valores se constroem e transformações ganham corpo. Elas impactam positivamente não apenas os alunos, mas também as instituições envolvidas e a comunidade ao redor, promovendo desenvolvimento sustentável. Por que essas parcerias são tão importantes?

- Podem ser um instrumento de fortalecimento da indústria nacional: a partir da aplicação de processo, algumas contribuições podem emergir: i) formalização de acordos e cooperação; ii) descrição do processo; implementação de práticas e sistemas; iii) transferência de tecnologias, iv) discussão dos benefícios entre os parceiros envolvidos (Nardi, *et al,* 2023).
- Promove a atualização constante e inserção no mercado de trabalho: projetos como este ampliam as chances de os estudantes conseguirem estágios e empregos em áreas alinhadas com as demandas tecnológicas atuais (Rego, Rosas e Prado, 2021).

- Fomenta à inovação e à pesquisa: a interface academia e indústria potencializa o desenvolvimento de soluções criativas e aplicadas aos desafios cotidianos. Essa prática já é conhecida como "habitat de inovação" (Spolidoro, 1997; Lima, 2015).
- Desenvolvem habilidades socioemocionais: além das competências técnicas, os alunos são estimulados a trabalhar em equipe, comunicar-se de forma eficiente e adaptar-se a diferentes contextos, habilidades essenciais para o mundo atual (Berlingeri, 2018).

Com efeito, é importante enfatizar que a interação entre academia e indústria pode fluir, ainda mais, quando a troca de experiência e de conhecimentos se dá quando os atores envolvidos compartilham semelhanças ou condutas próximas (Balconi, *et al,* 2004), ou seja, existam canais de agregação e fortalecimento. Sendo assim, é importante destacar que o curso foi possível graças à articulação de diferentes instituições que possuem entre si uma sintonia: a atuação no contexto amazônico.

1.4. INSTITUTO FEDERAL DO AMAZONAS (IFAM)

Como parte da Rede Federal de Educação Profissional e Tecnológica, o IFAM (Figura 1.6) tem como missão promover o desenvolvimento sustentável da Amazônia por meio da educação, ciência e tecnologia. É uma instituição comprometida com a inclusão, a excelência educacional e a valorização da diversidade, conforme descrição em sua página oficial (https://www2.ifam.edu.br/).

Figura 1.6 - Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – IFAM. Fachada do Campus Manaus Centro

Fonte: IFAM (2025).

1.5. LG *ELECTRONICS* DO BRASIL

Multinacional de renome, a LG (Life's Good) atua no Polo Industrial de Manaus na fabricação de produtos eletrônicos (Figura 1.7). A empresa acredita na inovação como caminho para garantir a satisfação de seus consumidores e entende a formação de profissionais qualificados como parte desse processo.

Figura 1.7 – Fábrica da LG Electronics em Manaus

Fonte: LG (2025).

1.6. FAEPI – FUNDAÇÃO DE APOIO DO IFAM

A Fundação de Apoio ao Ensino, Pesquisa, Extensão e Interiorização do IFAM (FAEPI) é uma instituição privada que atua no suporte à execução de projetos da instituição, contribuindo com o desenvolvimento acadêmico e científico. A Figura 1.8 apresenta uma foto da FAEPI.

Figura 1.8 - Foto da FAEPI

Fonte: https://faepi.org/sobre-nos/ (2025)

1.7. POLO DE INOVAÇÃO DO IFAM MANAUS

O INOVA é um espaço voltado à pesquisa aplicada e à inovação tecnológica. Com foco em controle e processos industriais, o polo tem papel central na articulação entre conhecimento científico e aplicação prática, especialmente dentro do contexto amazônico. A Figura 1.9 apresenta uma foto do INOVA.

Figura 1.9 - Foto do INOVA

Fonte: https://inova.ifam.edu.br/ (2025).

1.8. CONSIDERAÇÕES IMPORTANTES

A conexão entre indústria e academia, conforme se observa, é fundamental. Essa articulação na Amazônia, contexto marcado por complexidades, torna o processo ainda mais interessante. Os inúmeros desafios que caracterizam a região, como a biodiversidade, a logística e a multiplicidade de grupos étnicos, concorrem para a iminente necessidade de formar profissionais qualificados.

Sendo assim, o desenvolvimento do curso de robótica industrial na Amazônia representa uma ação estratégica no que diz respeito ao acesso às tecnologias emergentes. A estruturação curricular bem como sua infraestrutura oferta uma formação que contribui para o fortalecimento da EPT junto às tecnologias emergentes bem como para a valorização do potencial humano local. A Robótica Industrial, nesse contexto, emerge como vetor de divulgação da ciência, da tecnologia e da engenharia, ao passo que envolve e estimula os estudantes para a autonomia.

Outro elemento importante é que ao propor desafios que envolvam o uso da robótica a partir de desafios assentados no cotidiano da indústria, fomenta-se a elaboração de soluções tecnológicas contextualizadas. O conhecimento partilhado e produzido emerge de uma tácita realidade, onde situações reais dialogam com resolutividades pensadas e aplicadas em uma dada realidade. Essa perspectiva enaltece um processo material e criativo, elementos que estão articulados aos princípios da EPT que busca sintonizar seu fazer educativo com os arranjos produtivos locais.

Assim, a realização do curso, dessa maneira, reverbera a dimensão técnica, isto é, saber fazer, mas, essencialmente, deve projetar o elemento ético, onde a conexão é também com a realidade local e suas dinâmicas socioambientais. Iniciativas como essa demonstram que, quando a educação e a indústria se integram, todos ganham: os alunos, as instituições e a sociedade. Promover e investir em uma formação que envolva a robótica industrial em nossa região não é apenas uma ação educativa, mas reveste-se de um compromisso com a equidade de oportunidades aos estudantes da Amazônia.

REFERÊNCIAS

[1] AKDUR, Deniz. Analysis of software engineering skills gap in the industry. **ACM Transactions on Computing Education**, v. 23, n. 1, p. 1-28, 2022.

[2] BALCONI , M.; BRESCHI , S.; LISSONI , F. Networks of inventors and the role of academia: an exploration of Italian patent data**. Research Policy**, v. 33, n. 1, p. 127-145, Jan. 2004.

[3] BERLINGERI, Matheus Mascioli. Competências socioemocionais e mercado de trabalho: um estudo para o caso brasileiro. **Dissertação de mestrado**. Universidade de São Paulo: Ribeirão Preto, 2018.

[4] BRASIL. **Lei nº 11.741, de 16 de julho de 2008**. https://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2008/Lei/L11741.htm. Acesso no dia 27 de julho de 2025.

[5] BRASIL. **Lei nº 9.394, de 20 de dezembro de 1996**. https://www.planalto.gov.br/CCIVIL_03/Leis/L9394.htm. Acesso no dia 27 de julho de 2025.

[6] Fundação de Apoio ao Ensino, Pesquisa, Extensão e Interiorização do IFAM – FAEPI. Disponível em: https://faepi.org/sobre-nos/. Acesso no dia 26 de julho de 2025.

[7] GARGALLO LÓPEZ, Bernardo et al. La competencia aprender a aprender. **Revista Española de Pedagogía**, v. 78, n. 276, p. 187-212, 2020.

[8] Instituto Federal do Amazonas - IFAM. Missão e Visão. Disponível em: http://www2.ifam.edu.br/instituicao/missao-e-visao. Acesso no dia 24 de janeiro de 2024a.

[9] LG Eletronics. A nossa visão. Disponível em: https://www.lg.com/pt/descubra-lg/a-nossa-visao/. Acesso no dia 24 de janeiro de 2024.

[10] LIMA, Sérgio Henrique de Oliveira. A cooperação de pesquisa e inovação entre universidade e indústria no Brasil na perspectiva de gestores acadêmicos de pesquisa. 2015. 215 f. **Dissertação (Mestrado)** – Universidade Federal do Ceará, Faculdade de Economia, Administração, Atuária e Contabilidade, Programa de Pós-Graduação em Administração e Controladoria, Fortaleza-CE, 2015.

[11] Martino, Mariluci Alves. A importância das parcerias na educação profissional. 2006. 221 f. **Dissertação (Mestrado em Educação: Currículo)** - Programa de Estudos Pós-Graduados em Educação: Currículo, Pontifícia Universidade Católica de São Paulo, São Paulo, 2006.

[12] NAPKIN. Napkin AI - The visual AI for business storytelling. Disponível https://www.napkin.ai/

[13] NARDI, J. C.; PULINI, I. C.; CARVALHO, V. A. de; LIBERATO, A. B.; SAQUETTO, T. C. Colaboração Indústria-Academia: Desenvolvimento e Implantação de um Sistema de Controle E Monitoramento Na Indústria De Confecção. **Boletim de Conjuntura (BOCA)**, Boa Vista, v. 16, n. 46, p. 618–643, 2023. DOI: 10.5281/zenodo.10056266. Disponível em: https://revista.ioles.com.br/boca/index.php/revista/article/view/2408. Acesso em: 5 ago. 2025.

[14] Polo de Inovação IFAM. Finalidade. Disponível em: https://inova.ifam.edu.br/quem-somos/. Acesso no dia 24 de janeiro de 2024.

[15] REGO, Fátima Aparecida; ROSAS, Iris Renata de Carvalho; PRADOS, Rosália Maria Netto. Educação Profissional e Tecnológica como alternativa de acesso ao mercado de trabalho. **Brazilian Journal of Development**, *7*(2), 14585–14596, 2021.

[16] SPOLIDORO, Roberto. A Sociedade do Conhecimento e Seus Impactos no Meio Urbano. In: PALADINO, Gina; MEDEIROS, Lucília Atas (org). **Parques Tecnológicos e Meio Urbano. Artigos e Debates**. Brasília: ANPROTEC, 1997.

CAPÍTULO

02 Disciplina de múltiplas competências e *hands on*

Ana Cláudia Ribeiro de Souza[1], José Pinheiro de Queiroz Neto[2]

2.1. CARACTERIZAÇÃO DA DISCIPLINA

Esta disciplina introduz os conceitos e métodos aplicados durante toda a formação no curso avançado em Robótica Industrial – 2.0, que serão utilizados nas demais disciplinas e na Jornada *Hands On*. Ancorada na teoria do construtivismo de Jean Piaget e materializada por John Dewey no que chamou de filosofia *Learning by doing* (aprender fazendo), esta formação utiliza o que chamaremos de filosofia *hands on* ao longo de todo o processo formativo dos discentes (Moran, 2018).

Esta disciplina, portanto, apresenta a convergência de duas Metodologias Ativas que, integradas, potencializam as múltiplas competências que os discentes já possuem, utilizando a integração das metodologias de Aprendizagem Baseada Competências (*Competence-Based Learning* – CBL), Aprendizagem Baseada em Projetos (*Project-Based Learning* – PBL) e *Design Thinking* (DT) para desenvolver competências na integração das habilidades cognitivas (*hard skills*) e socioemocionais (*soft skills*), em um ambiente acadêmico/industrial que traz uma educação significativa e potencializa a aprendizagem dos alunos.

2.1.1. EMENTA

A disciplina apresentou como ementa: Conceitos de Múltiplas Competências; Conceitos e aplicações de *Project Based Learning* (PBL); Conceitos e aplicações de Design Thinking (DT); Fases do DT: Imersão; Ideação; Prototipação. Aplicação prática.

2.1.2. OBJETIVOS

Múltiplas competências e *hands on*, teve como objetivos preparar os alunos para a compreensão e utilização da formação por competências em um ambiente *hands on* (aprender fazendo), e introduzir a metodologia *Design Thinking* (DT) de desenvolvimento de produtos e serviços focados nas necessidades, desejos e limitações dos usuários, em abordagens de soluções que possam converter dificuldades e limitações em benefícios para o cliente e valor de negócio para a sua empresa.

[1] Doutora em História Social, Docente da Disciplina de Múltiplas Competências e *Hands On*. Professora titular do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas– IFAM. E-mail: *ana.souza@ifam.edu.br*

[2] Doutor em Ciência da Computação. Pesquisador Educacional do Curso de Capacitação Avançada de Robótica Industrial. Professor titular do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – IFAM. E-mail: *pinheiro@ifam.edu.br*

2.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 24h

Carga Horária Prática: 16h

Total: 40h

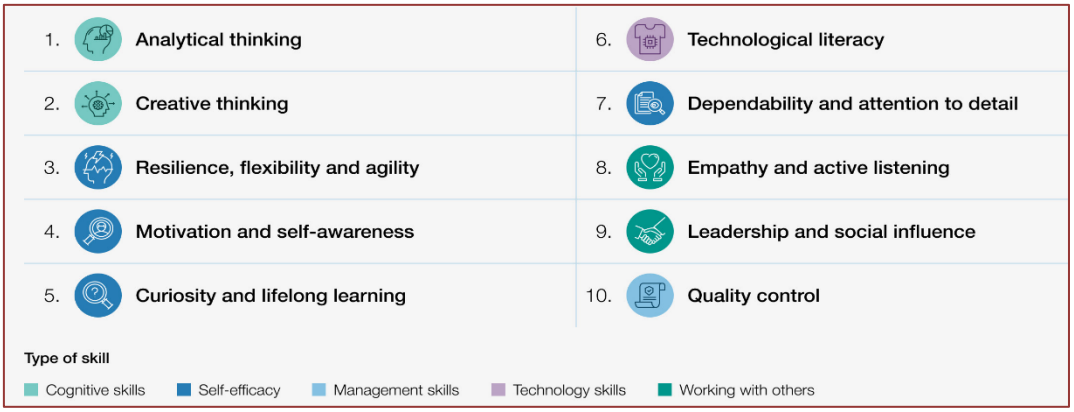
2.2. CONTEÚDOS

2.2.1. MÚLTIPLAS COMPETÊNCIAS

Diversas instituições acadêmicas têm repensado suas práticas de ensino para desenvolver não apenas habilidades técnicas (*hard skills*), mas também habilidades socioemocionais (*soft skills*) nos estudantes, que serão futuros profissionais no mundo do trabalho. As *soft skills*, que incluem liderança, comunicação, resolução de conflitos, empatia, ética, flexibilidade e gestão de equipes, entre outros, são vistas como complementares, mas não menos importantes. A ideia é alcançar uma formação integral, que considera não apenas aspectos cognitivos, mas também físicos, intelectuais, sociais, políticos e científicos. (Queiroz-Neto *et al*., 2021).

De fato, a formação para o mundo do trabalho não pode ser uma formação meramente tecnicista, superando a dualidade da divisão entre trabalho manual e trabalho intelectual. Conforme o Fórum Econômico Mundial (Fórum, 2023), as principais habilidades necessárias para o mundo do trabalho em 2027 podem ser sumarizadas na Figura 2.1. Observe que estas habilidades se dividem entre *soft* e *hard skills*, exigindo que o discente desenvolva diversificadas habilidades, ou competências, como preferimos chamar.

Figura 2.1 – Habilidades para o mundo do trabalho em 2027



Fonte: Fórum (2023).

Portanto, é necessário desenvolver nos discentes suas Múltiplas Competências, que são habilidades que permitem a uma pessoa desempenhar diferentes tarefas ou funções com eficiência e qualidade. As múltiplas competências podem ser adquiridas por meio de educação, treinamento, experiência ou autoaprendizagem. As múltiplas competências são valorizadas no mercado de trabalho, pois aumentam a flexibilidade, a adaptabilidade e a criatividade dos profissionais, e também podem contribuir para o

desenvolvimento pessoal e social, pois ampliam os horizontes e as possibilidades de atuação dos educandos.

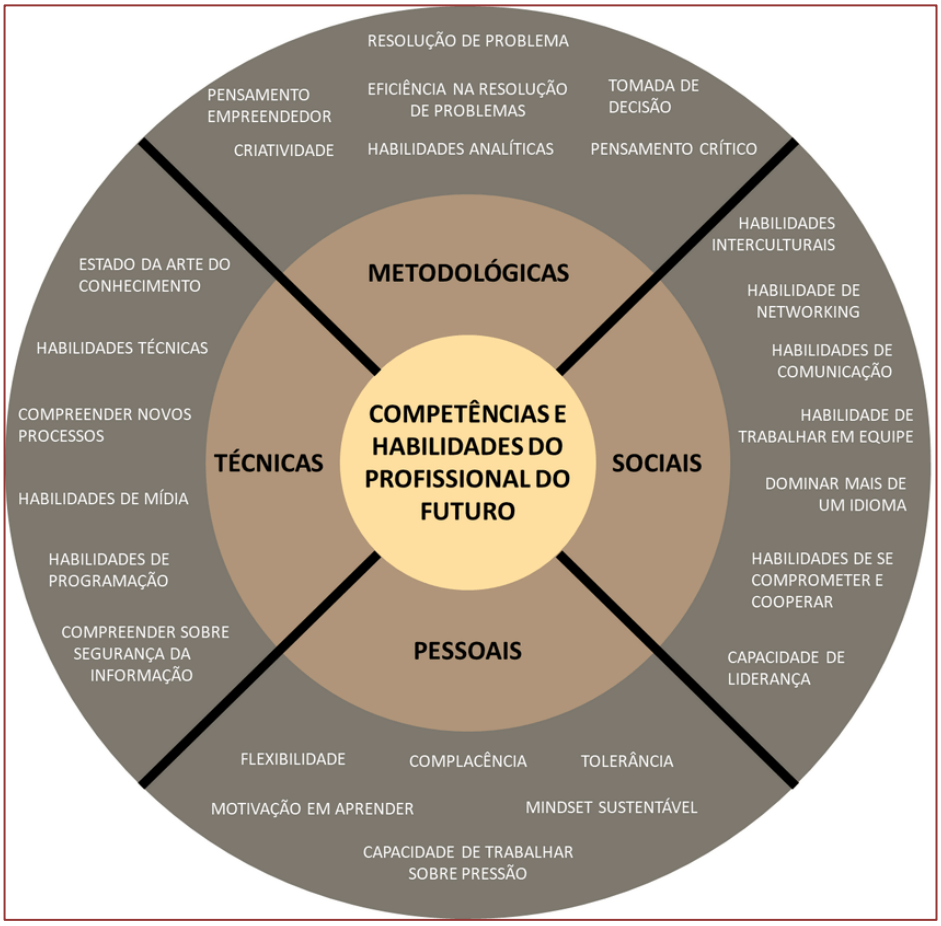
As múltiplas competências são, na prática, o ato de desenvolver nossas múltiplas inteligências. A teoria das múltiplas inteligências foi proposta pelo psicólogo e educador Howard Gardner, na década de 80, e defende a proposta de que existem diferentes tipos de inteligência que se manifestam em diferentes domínios da vida, seja no pessoal ou no profissional. Segundo Gardner (1995), as inteligências são: linguística, lógico-matemática, musical, espacial, corporal-cinestésica, interpessoal, intrapessoal e naturalista. Cada pessoa possui um perfil de inteligências que se desenvolve ao longo da vida, de acordo com as experiências e oportunidades de aprendizagem. Reconhecer as múltiplas inteligências é importante para valorizar a diversidade humana e promover o desenvolvimento integral dos indivíduos, conforme ilustra a Figura 2.2

Figura 2.2 – Múltiplas Inteligências do ser humano

Fonte: Gardner (1995).

Portanto, utilizar estas múltiplas inteligências para gerar múltiplas competências é um desafio importante quando se trata de formação de alto nível, para a qual se faz necessária uma metodologia que possa entender estes aspectos e explorá-los de maneira proativa. A Figura 2.3 apresenta habilidades e competências que, segundo Hecklau, Orth, Kidschun e Kohl seriam do futuro, mas que são atuais e cada vez mais necessárias (Hecklau *et al.*, 2017).

Figura 2.3 – Múltiplas Inteligências do ser humano



Fonte: Hecklau *et al.*, (2017).

2.2.2. O MÉTODO CBL/PBL/DT

A formação na disciplina Múltiplas Competências e *Hands On* empregou a proposta de um novo modelo educacional integrado, que combina três metodologias importantes: abordagens baseadas em Competências, em Projetos e em *Design Thinking*. A Aprendizagem Baseada em Competências (*Competence-Based Learning – CBL*) estabelece os fundamentos ao garantir que os alunos adquiram o conhecimento técnico essencial e as habilidades necessárias para operar sistemas robóticos sofisticados (Henri *et al.*, 2017).

Concomitante a isso, a Aprendizagem Baseada em Projetos (*Project-Based Learning – PBL*) enfatiza o aprendizado por meio de desafios do mundo real, incentivando os estudantes a aplicar conceitos teóricos em ambientes práticos e colaborativos (Almulla, 2020).

Por fim, o método *Design Thinking* introduz um processo iterativo e centrado no usuário, que estimula a criatividade e a inovação, permitindo que os alunos desenvolvam soluções inéditas adaptadas às demandas em constante evolução da Indústria 4.0 (Panke, 2019).

Ao sintetizar essas metodologias, o modelo proposto busca aproximar a teoria da prática, preparando uma força de trabalho completa e capaz de impulsionar a inovação

digital e a eficiência na robótica industrial. Essa estratégia abrangente não apenas atende aos requisitos técnicos da manufatura moderna, mas também desenvolve habilidades interpessoais e atitudes inovadoras indispensáveis para construir uma indústria resiliente e preparada para o futuro, que já é, de fato, o nosso presente.

Esta combinação de metodologias resulta em um processo de aprendizagem que promove a autonomia dos alunos, exigindo do professor conhecimento e disposição para atuar como mentor. O método coloca o aluno no centro do processo, permitindo-lhe agir de forma autônoma e visando uma formação humana integral (Barbosa; Moura, 2013).

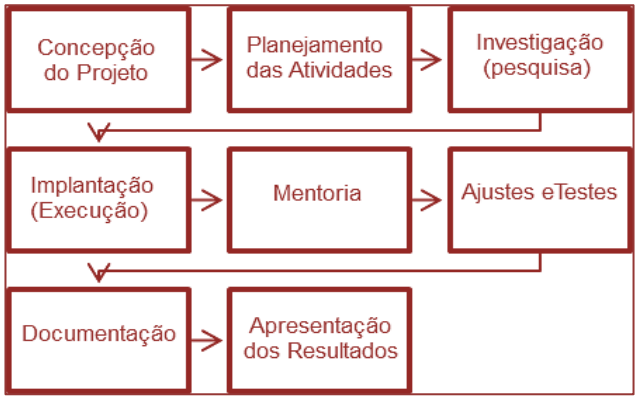
No decorrer da formação os discentes das turmas foram divididos em equipes, e as interações colaborativas das equipes, formadas por integrantes multidisciplinares, proporcionaram o ambiente ideal para o desenvolvimento das competências necessárias às aplicações de Robótica Industrial de alto nível. Entre as habilidades aprimoradas, destacam-se a liderança, a comunicação, a resolução de conflitos, a empatia, a ética, a flexibilidade e a gestão de equipes.

A PBL é um método de ensino que envolve os alunos em projetos do mundo real, significativos para eles. Os alunos, divididos em equipes, trabalham em um projeto por um período definido, resolvendo um problema do mundo real ou respondendo a uma pergunta complexa. Eles demonstram seus conhecimentos e habilidades criando um serviço ou produto para um público real, neste caso, para a empresa parceira do projeto (Almulla, 2020).

Importante ressaltar que PBL difere de somente "fazer um projeto", pois exige pensamento crítico, resolução de problemas, colaboração e várias formas de comunicação. Os alunos precisam fazer mais do que apenas lembrar informações.

Ao utilizar a PBL, os alunos precisam usar habilidades de pensamento para buscar, interpretar, utilizar e transformar essas informações, além de aprender a trabalhar em equipe. (Torre-Neches, 2020). Diversos autores apresentam um fluxo de etapas para aplicação da PBL, e neste projeto utilizamos o desenvolvido por Queiroz-Neto *et al.* (2015) e dos Santos et al. (2019), apresentado na Figura 2.4.

Figura 2.4 – Etapas da PBL



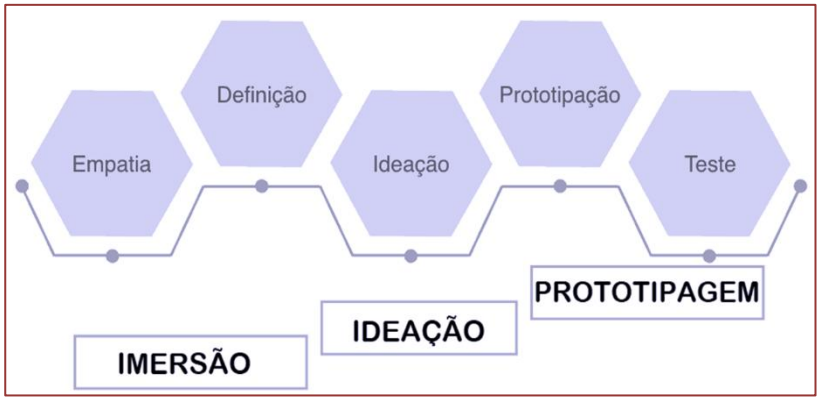
Fonte: Traduzido de Queiroz-Neto *et al.*, (2015).

O *Design Thinking* (DT) é um método que visa resolver problemas de maneira criativa, levando em consideração as necessidades dos usuários e seus feedbacks na criação de produtos ou serviços. Ele envolve a observação de contextos da vida real, a análise de dados para entender melhor as necessidades do usuário e a adoção de uma mentalidade criativa para formar ideias e testar soluções. O DT não é um método específico, mas uma abordagem que organiza ideias para estimular a tomada de decisões e a busca pelo conhecimento, que classicamente possui cinco etapas: empatizar, definir, idealizar, prototipar e testar (Panke, 2019).

Nesta formação utilizamos uma versão prática do DT com três etapas que contém as cinco etapas originais (Figura 2.5) que são: imersão (empatia e definições), ideação e prototipação (incluindo testes). A imersão é a forma de estudar o problema e compreendê-lo sobre diversas perspectivas e definir os limites do projeto, criando-se um entendimento das necessidades a serem sanadas.

A Ideação é a geração de propostas de soluções direcionadas por critérios norteadores, oriundos da imersão, de forma a atender as necessidades mapeadas dos clientes e responder ao problema definido. Finalmente, a Prototipação tem como objetivo testar de maneira rápida, com baixo custo, a solução encontrada e submetê-la para aprovação do cliente (Queiroz-Neto *et al.*, 2021).

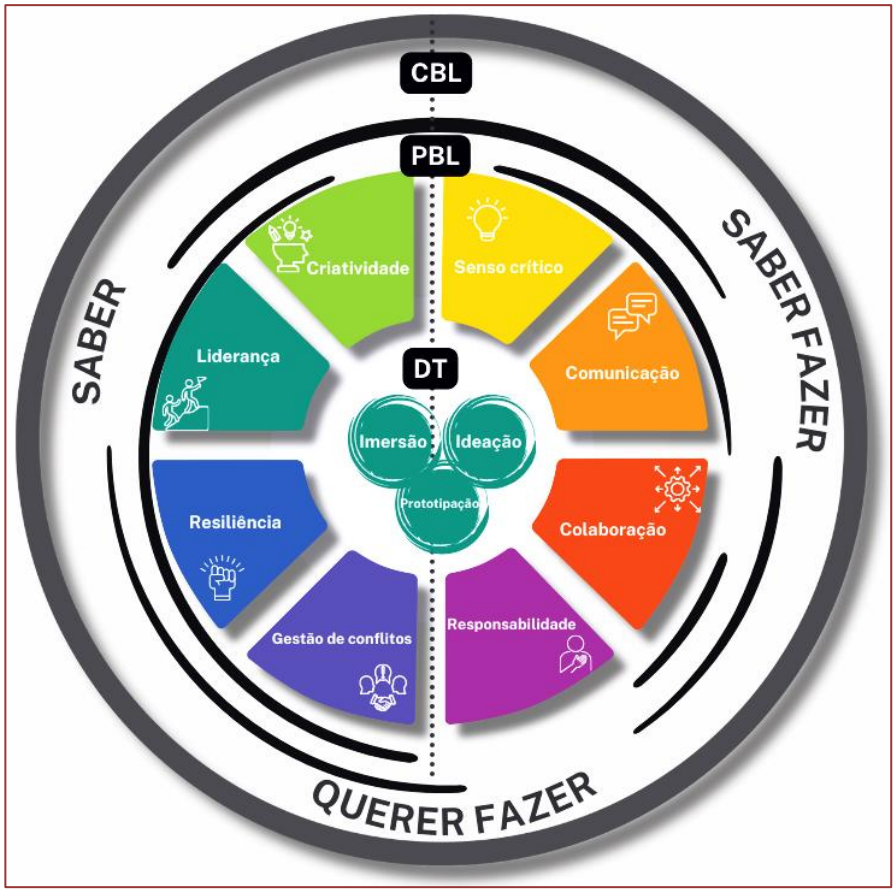
Figura 2.5 – Etapas do *Design Thinking*



Fonte: Os autores, 2025.

A Figura 2.6 ilustra a metodologia proposta na formação, baseada em CBL/PBL/DT. Ela integra três abordagens inovadoras para um aprendizado mais holístico e eficaz. Primeiro, emprega o *Design Thinking*, atualmente utilizado em empresas inovadoras, que estimula uma abordagem criativa para gerar processos e produtos inovadores. Os estudantes aprendem a identificar as necessidades dos usuários, desenvolver soluções inovadoras, criar protótipos e testar essas ideias de forma iterativa.

Figura 2.6 – CBL/PBL/DT integrados para a formação em Robótica Industrial



Fonte: Os autores, 2025.

Em seguida, incorpora a PBL, envolvendo os estudantes em projetos práticos e do mundo real, que lhes permitem aplicar diretamente seus conhecimentos na resolução de problemas autênticos, proporcionando desenvolver as *hard skills* (habilidades cognitivas) e *soft skills* (habilidades socioemocionais).

Por fim, integra a CBL, que enfatiza a demonstração de competências: os alunos progridem em seu próprio ritmo, avançando apenas quando dominam o conteúdo, conforme medido por avaliações formativas. Essa metodologia é projetada para cultivar não apenas conhecimento e habilidades, mas também os aspectos atitudinais apropriados necessários para a competência plena, ou seja: o SABER, que representa o domínio do conhecimento teórico e conceitual; o SABER FAZER, que se manifesta na aplicação prática e eficiente dos conhecimentos adquiridos; e o QUERER FAZER, que integra as habilidades socioemocionais e a motivação pessoal, refletindo uma postura proativa, ética e colaborativa diante dos desafios profissionais.

Ao integrar essas metodologias, promovemos um ambiente de aprendizagem dinâmico, onde conceitos teóricos são continuamente aplicados a desafios do mundo real. Além disso, ao reforçar o processo de aprendizagem por meio do desenvolvimento de competências essenciais, essa abordagem multifacetada não apenas estimula a compreensão profunda do conteúdo, como também capacita os estudantes com habilidades práticas necessárias para projetar e executar projetos bem-sucedidos

Ao longo da disciplina um projeto prático foi desenvolvido. A PBL definiu a organização geral do projeto, incluindo suas etapas macro, o conceito de mentoria, o foco no estudante e sua autonomia, e a avaliação formativa, através de orientações regulares e acompanhamento dos times de alunos. A DT, por sua vez, trouxe a organização interna do projeto, definindo etapas dentro do desenvolvimento do projeto em si, com base na tríade imersão, ideação e prototipação.

Com isto, entendemos que para além da aprendizagem em Robótica Industrial, esta metodologia permitiu ao aluno uma formação humana integral, sendo o entendimento de que esta gira em torno da integração de todas as dimensões e potencialidades do indivíduo no processo educativo. Essas dimensões, segundo Frigotto (2012, p. 267) "envolvem sua vida corpórea material e seu desenvolvimento intelectual, cultural, educacional, psicossocial, afetivo, estético e lúdico".

2.3. PROCESSO AVALIATIVO

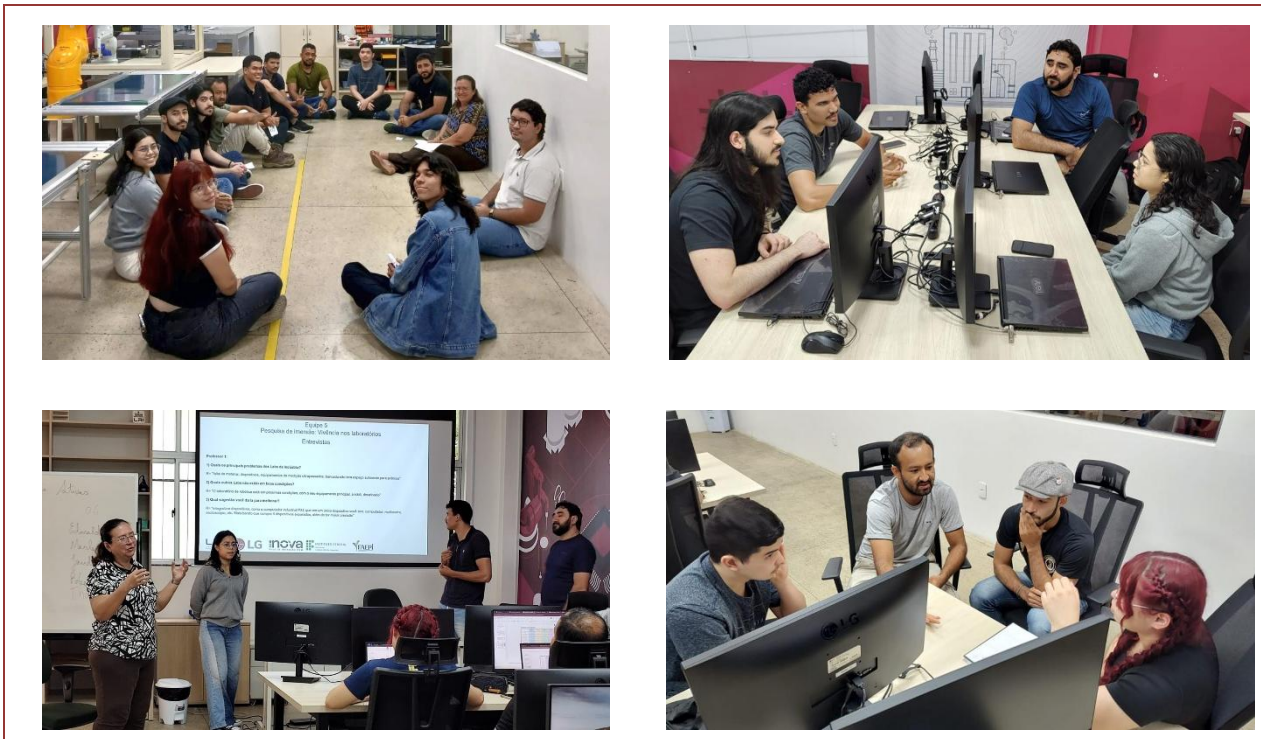
Nesta disciplina, foi utilizada a avaliação formativa, que tem seu foco no processo ensino-aprendizagem e está diretamente interligada a este, constituindo-se parte de um todo e não seu fim em si, e não tem finalidade de medição da aprendizagem em parâmetros quantitativos, mas assume um caráter objetivamente pedagógico sobre o que se aprendeu, como foco maior no processo (Barreira, 2019).

Com isto, a avaliação se deu a partir de experimentação prática da filosofia *hands on* e utilização da metodologia PBL/DT, na qual as duas turmas foram divididas em seis times multidisciplinares, ou seja, diversificando a base de formação dos alunos, e estes desenvolveram as etapas do método para apresentar uma solução a um problema real apresentado no Campus Manaus Distrito Industrial, seguindo as etapas da imersão, ideação e prototipagem. Os problemas foram de livre escolha dos times, mas delineados nas seguintes áreas: Portaria/entrada, Limpeza, Ambientes Acadêmicos.

A Figura 2.7 apresenta quatro momentos da disciplina: integração dos alunos para autoconhecimento e conhecimento; discussão em equipe dos dados obtidos na imersão; uso de brainstorming para a fase da ideação e desenvolvimento do protótipo.

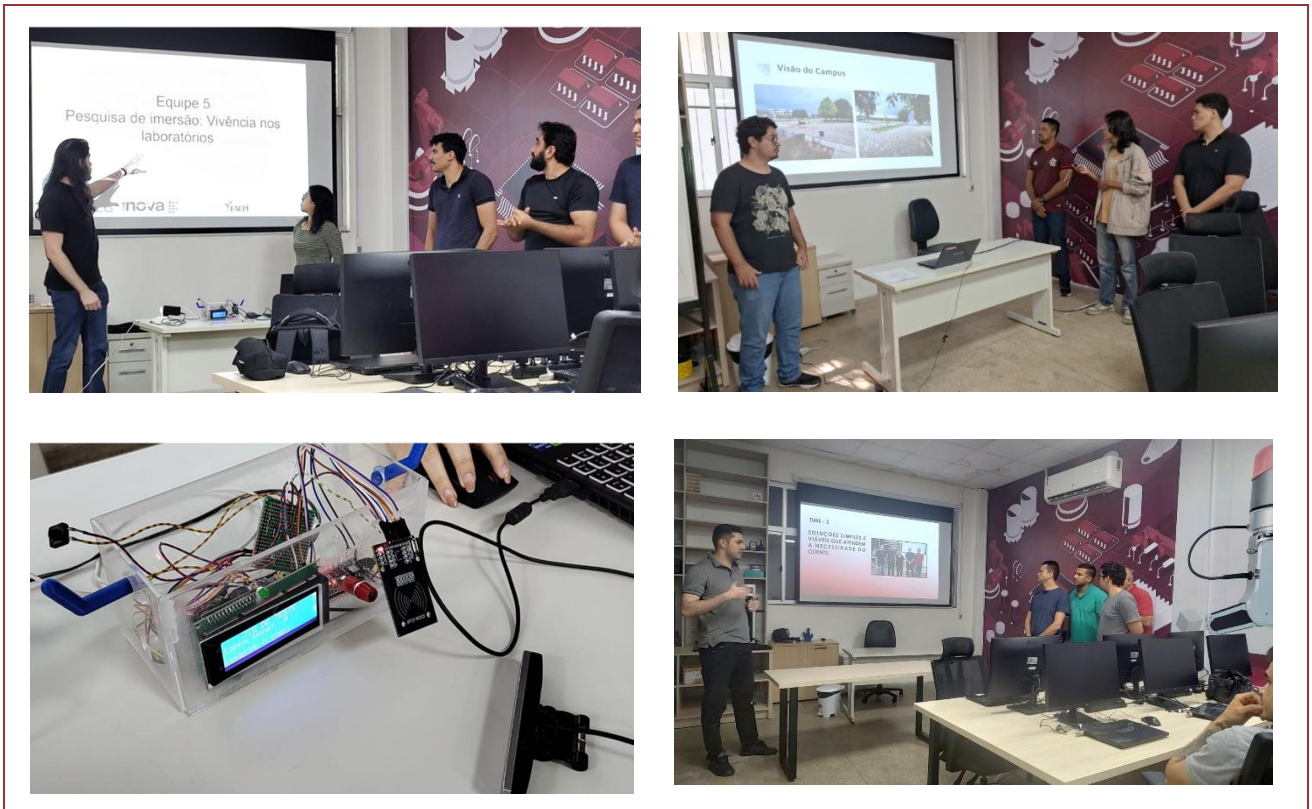
A Figura 2.8 apresenta a fase final da avaliação formativa, com a apresentação do desenvolvimento prático das fases do PBL/DT e dos resultados obtidos aos professores da disciplina e convidados, a saber: o diretor do Polo de Inovação e o Professor Orientador da jornada *hands on* do curso. Ao final da disciplina, não houve evasão de nenhum aluno e todos foram aprovados, com eficiência de 100% na disciplina.

Figura 2.7 – Momentos da formação usando CBL/PBL/DT na prática



Fonte: Os autores, 2025

Figura 2.8 – Apresentação do desenvolvimento da prática e resultados obtidos



Fonte: Os autores, 2025

REFERÊNCIAS

[1] ALMULLA, M. A., "The Effectiveness of the Project-Based Learning (PBL) Approach as a Way to Engage Students in Learning", **SAGE Open**, vol. 10. N. 3, July 2020.

[2] BARBOSA, E.F; MOURA, D.G. Metodologias ativas de aprendizagem na Educação Profissional e Tecnológica. **Tec. Senac**, Rio de Janeiro, v. 39, n.2, p.48-67, maio/ago. 2013. <Disponível em: http://www.bts.senac.br/index.php/bts/article/view/349>. Acesso em: 02 mai. 2019.

[3] BARREIRA, C. Concepções e práticas de avaliação formativa e sua relação com os processos de ensino e aprendizagem. In book: **Avaliar para aprender no Brasil e em Portugal: perspectivas teóricas, práticas e de desenvolvimento** (pp.191-218). Editora CRV Ltda. DOI: 10.248224/978854443463.5. Setembro, 2019.

[4] FORUM Econômico Mundial. The future of job reports 2023. <Disponível em: https://www.weforum.org/publications/the-future-of-jobs-report-2023/>. Acesso em 03 nov. 2023.

[5] FRIGOTTO, G. Educação omnilateral. In: CALDART, R. S. et al. **Dicionário da Educação do Campo**. Rio de Janeiro, São Paulo: Expressão Popular, 2012.

[6] GARDNER, H. "**Estruturas da mente – A Teoria das Inteligências Múltiplas**". 1995

[7] HECKLAU, F., ORTH, R., KIDSCHUN, F., KOHL, H. **Human Resources Management**: Meta-Study - Analysis of Future Competences in Industry 4.0., 2017.

[8] HENRI, M., JOHNSON, M. D., e NEPAL, B. "A Review of Competency-Based Learning: Tools, Assessments, and Recommendations". **J. Eng. Educ**., 106: 607-638, 2017. doi: 10.1002/jee.20180.

[9] MORAN, J. (org.). **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Porto Alegre: Penso, 2018.

[10] PANKE, Stefanie. "Design Thinking in Education: Perspectives, Opportunities and Challenges" **Open Education Studies**, vol. 1, no. 1, 2019, pp. 281-306. https://doi.org/10.1515/edu-2019-0022.

[11] PIAGET, Jean. Psicologia e pedagogia. Tradução de Dirceu Accioly Lindoso e Rosa Maria Ribeiro da Silva. São Paulo e Rio de Janeiro: Editora Forense, 1970.

[12] QUEIROZ-NETO, J. P., SALES, D. C., PINHEIRO, H. S., NETO, B. O., "Using modern pedagogical tools to improve learning in technological contents", 2015 IEEE **Frontiers in Education Conference (FIE)**, El Paso, TX, 2015, pp. 1-8, doi: 10.1109/FIE.2015.7344383.

[13] QUEIROZ-NETO, J. P.; FARIAS, M. S. F.; CHAGAS, E. L. T., "Project Based Learning e Design Thinking em um projeto de intercâmbio". **Revista Ibero-Americana de Estudos em Educação**, Araraquara, v. 16, n. 3, p. 1791-1806, jul./set. 2021. e-ISSN: 1982-5587. DOI: https://doi.org/10.21723/riaee.v16i3.14557

[14] QUEIROZ-NETO, J. P.; CHAGAS, E. L. T.; RAMOS, C. . **LAPASSION Manaus: uma experiência internacional de formação profissional de alto nível**. 1. ed. Curitiba: Appris, 2021. v. 1. 117p .

[15] SANTOS, J. M.; BREMGARTNER, V.; QUEIROZ-NETO, J. P.; LIMA, H.; PEREIRA, M. "ROBÔ-TI: Educational Robotics and Project-Based Learning Stimulating High School Students in the Information Technology Area," 2019 **IEEE Frontiers in Education Conference (FIE)**, Covington, KY, USA, 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028401.

[16] TORRE-NECHES, B. De La, RUBIA-AVI, M., M., APARICIO-HERGUEDAS, J. L. "Project-based learning: an analysis of cooperation and evaluation as the axes of its dynamic". **Humanit Soc Sci Commun** 7, 167 (2020). https://doi.org/10.1057/s41599-020-00663-z.

CAPÍTULO

03 Manufatura aditiva

Renan Cavalcante Santos[1], Davi da Silva Ribeiro Castro[2]

3.1. CARACTERIZAÇÃO DA DISCIPLINA

A disciplina de Manufatura Aditiva foi concebida para integrar teoria e prática em um contexto laboratorial, utilizando metodologias ativas que combinam aulas expositivas dialogadas, demonstrações práticas, atividades em grupo e resolução de problemas técnicos. Como recursos pedagógicos, são utilizados computadores com licenças de software de modelagem assistida por computador (CAD), impressoras 3D, materiais de impressão (como filamentos à base de PLA), projetores multimídia, bancadas de montagem e softwares de fatiamento.

Durante o curso, são abordados conteúdos fundamentais para o entendimento e a aplicação prática da manufatura aditiva, uma tecnologia emergente e estratégica para a transformação da indústria.

Os alunos trabalham os conceitos fundamentais de desenho auxiliado por computador (CAD) aplicados ao projeto de componentes mecânicos, abrangendo desde as etapas iniciais de concepção até a elaboração da documentação técnica final. Na sequência, são apresentados de forma detalhada os princípios e fundamentos da manufatura aditiva, abordando suas diferentes tecnologias, materiais disponíveis e parâmetros de processo, bem como as principais vantagens, limitações e critérios de seleção desse método quando comparado a processos convencionais de fabricação mecânica. Também são discutidas a seleção e aplicação de materiais, práticas de segurança operacional e noções de pós-processamento.

A disciplina também enfatiza o desenvolvimento de projetos práticos, que apresentam soluções funcionais para necessidades reais num contexto industrial. Um dos desafios propostos consiste na criação de garras manipuladoras para manuseio de peças, integrando conhecimentos de projeto mecânico, manufatura aditiva e análise funcional. Essa abordagem prática visa não apenas o domínio técnico, mas também o desenvolvimento de habilidades criativas, resolução de problemas e trabalho em equipe, reforçando a formação profissional e a capacidade de inovação dos participantes.

[1] Mestre em Engenharia Mecânica. Docente da Disciplina de Manufatura Aditiva. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM. E-mail: *renan.cavalcante@ifam.edu.br*

[2] Mestre em Engenharia Elétrica. Docente da Disciplina de Manufatura Aditiva. Engenheiro Instrutor do Flextronics Instituto de Tecnologia - FIT. E-mail: *davisrcastro@gmail.com*

3.1.1. EMENTA

A disciplina tem início com a introdução ao desenho auxiliado por computador (CAD), abordando os fundamentos de modelagem 3D, interface de software, criação e edição de esboços (*sketch*), operações de extrusão e revolução, bem como conceitos básicos de projeto mecânico e montagem de componentes.

Em seguida, são explorados os princípios da manufatura aditiva, incluindo seu histórico, principais tecnologias e comparação com outros processos de fabricação mecânica, o foco recai sobre a tecnologia Fused Deposition Modeling (FDM).

Os principais tópicos abordados ao longo da disciplina são:

1. Desenho auxiliado por computador aplicado a projeto mecânico: modelagem paramétrica, desenvolvimento de componentes mecânicos e documentação técnica.
2. Fundamentos de manufatura aditiva, histórico, tecnologias e aplicações;
3. Processo FDM: fundamentos, parâmetros de processo e materiais.
4. Preparação para Impressão 3D: fatiamento, configuração de parâmetros e otimização.
5. Operação e Manutenção de Impressoras 3D: procedimentos de operação segura, calibração e manutenção preventiva.

O conteúdo é aplicado em um projeto prático integrador, no qual os alunos desenvolvem e fabricam, por impressão 3D, garras manipuladoras para manuseio de peças. O projeto envolve desde o planejamento e modelagem CAD, passando pela preparação para impressão, até a análise funcional dos protótipos produzidos, incentivando a aplicação prática dos conceitos, a resolução de problemas e o desenvolvimento de competências criativas e técnicas.

3.1.2. OBJETIVOS

Desenvolver competências técnicas e práticas em projeto mecânico assistido por computador e manufatura aditiva, capacitando os alunos a conceber, modelar e fabricar componentes funcionais. A disciplina busca aprimorar habilidades de interpretação de requisitos, resolução de problemas, seleção de tecnologias e parâmetros de processo, bem como estimular a criatividade, a inovação e o trabalho colaborativo na execução de projetos aplicados.

3.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 15h

Carga Horária Prática: 25h

Total: 40h

3.2. CONTEÚDOS

Esta seção apresenta os principais tópicos que compõem o conteúdo programático.

3.2.1. DESENHO AUXILIADO POR COMPUTADOR (CAD)

Os sistemas Computer-Aided Design (CAD) representam a base tecnológica fundamental para a transformação digital nos processos de manufatura contemporâneos. Conforme demonstrado por Ustundag e Cevikcan (2018), a quarta revolução industrial promoveu uma evolução significativa no papel dos sistemas CAD, transformando-os de meras ferramentas de desenho assistido para componentes essenciais do fluxo digital integrado que caracteriza a Indústria 4.0. Os arquivos CAD modernos incorporam não apenas informações geométricas tridimensionais, mas também dados críticos sobre materiais, tolerâncias dimensionais e propriedades físicas dos componentes, constituindo assim modelos digitais completos que servem como base para todo o processo produtivo (Beyca et al., 2018).

No contexto específico da manufatura aditiva, a relação entre os sistemas CAD e os processos de fabricação assume características particularmente simbióticas. Como destacam Beyca et al. (2018), todo o processo de fabricação aditiva necessariamente se inicia com um modelo CAD, que posteriormente é convertido em instruções específicas de camadas para as impressoras 3D. Esta integração direta entre projeto e fabricação possibilita três vantagens principais: primeiro, permite a prototipagem rápida com ciclos iterativos de design significativamente mais ágeis; segundo, viabiliza a produção de geometrias complexas que seriam impossíveis ou extremamente difíceis de obter através de métodos tradicionais de usinagem; e terceiro, possibilita a personalização em massa através da parametrização eficiente de projetos (Beyca et al., 2018).

3.2.2. EVOLUÇÃO HISTÓRICA DA MANUFATURA ADITIVA

A década de 80 e início da década de 90 foram marcantes para o desenvolvimento de tecnologias que servem de base para o que conhecemos hoje como manufatura aditiva. Podemos citar como marco inicial o ano de 1981 com a publicação do primeiro conceito de prototipagem rápida desenvolvido pelo inventor Hideo Kodama que utilizava uma técnica de solidificação por luz ultravioleta. Em 1984 o processo, até então conhecido como impressão 3D, tornou-se economicamente viável quando o engenheiro Chuck Hull desenvolveu a tecnologia de Estereolitografia (SLA), que utiliza um laser ultravioleta para solidificar resina líquida camada por camada.

No final da década de 1980, surgiram outras tecnologias importantes: em 1988, Scott Crump inventou o processo *Fused Deposition Modeling* (FDM), que funde e deposita termoplásticos em camadas para construir a peça, processo hoje amplamente utilizado por sua acessibilidade e versatilidade. Em 1989, foi desenvolvido o processo *Selective Laser Sintering* (SLS) por Carl Deckard na Universidade do Texas, que utiliza laser para sinterizar pós plásticos ou metálicos, permitindo produção de peças funcionais com maior resistência. Já no início da década de 90 a empresa DTM (*Digital Techniques Manufacturing*) lançou o processo *Direct Metal Laser Sintering* (DMLS), uma evolução do SLS aplicada a metais, possibilitando a fabricação aditiva de componentes metálicos complexos para indústria aeroespacial e médica.

Desde então, diversas outras tecnologias e aprimoramentos surgiram, onde o foco principal tornou-se melhorar parâmetros de processos, aumentar qualidade e

produtividade dos componentes fabricados. Desta forma o processo deixou de servir apenas para fabricar protótipos e passou a produzir peças e componentes mecânicos acabados, com isso o processo de fabricação anteriormente conhecido como impressão 3D ou prototipagem rápida precisou evoluir sua definição e passou a ser conhecido como manufatura aditiva.

Dessa forma, a partir de 2009-2010, houve um esforço da comunidade científica na direção de melhor definir e caracterizar esse princípio de adição como um importante meio de fabricação. Assim, a denominação que melhor o descreve e que se consolidou na academia e na indústria (inclusive com normas técnicas – ver Seção 1.5) é a de manufatura aditiva. (Volpato,2021).

3.2.3. FUNDAMENTOS DE MANUFATURA ADITIVA

Segundo Volpato (2021), a manufatura aditiva é um processo de fabricação baseado na adição sucessiva de camadas de material a partir de um modelo geométrico tridimensional (3D). Trata-se de um método altamente automatizado, com grande potencial para produzir peças de geometrias complexas e elevado grau de personalização. Já Rodrigues (2017) reforça que este processo confere flexibilidade para a produção em baixos lotes e versatilidade de materiais e geometrias, sendo, portanto, uma tecnologia capaz de revolucionar processos produtivos e de alterar estratégias de produção atualmente empregadas, o que explica a crescente aplicação em diversas áreas e cadeias produtivas.

O uso da manufatura aditiva no Brasil tem o potencial de transformar significativamente o desenvolvimento de produtos e tecnologias em diversos setores, como saúde, construção civil, educação e tecnologia assistiva, estudos revelam uma tendência crescente no cenário brasileiro de utilizar a impressão 3D como uma ferramenta estratégica para o desenvolvimento de tecnologias industriais, com foco em modernização de processos, redução de custos e personalização de produtos (de Paula Netto,2024).

O processo inicia-se com a criação de um modelo 3D digital, normalmente desenvolvido em softwares CAD, no qual são especificadas as dimensões, formas, tolerâncias e demais detalhes do projeto. Esse modelo é então convertido para um formato compatível com a impressora 3D — sendo o STL (*Standard Tessellation Language*) o mais utilizado, devido à sua ampla compatibilidade com diferentes marcas de impressoras e softwares de fatiamento.

Na etapa de fatiamento (*slicing*), um software específico divide o modelo em camadas bidimensionais (“fatias”), que serão fabricadas de forma sequencial. Nesse momento, são definidos parâmetros de processo que influenciam diretamente a qualidade final da peça, como:

- Espessura da camada;
- Velocidade de deposição;
- Estratégia de preenchimento interno (*infill*);
- Geração de suportes para geometrias complexas.

Antes de iniciar a fabricação, é necessário realizar a preparação da impressora. Isso inclui:

- Verificação mecânica (alinhamento de eixos, calibração da mesa e dos bicos);
- Configuração de temperatura de mesa e extrusores;
- Preparação e carregamento do material (filamento, resina, pós metálicos ou outros, conforme a tecnologia utilizada).

Com o equipamento pronto, inicia-se a fabricação camada a camada, seguindo as instruções geradas no fatiamento, iniciando na base até atingir o topo, até que a peça atinja a geometria e as dimensões especificadas no projeto.

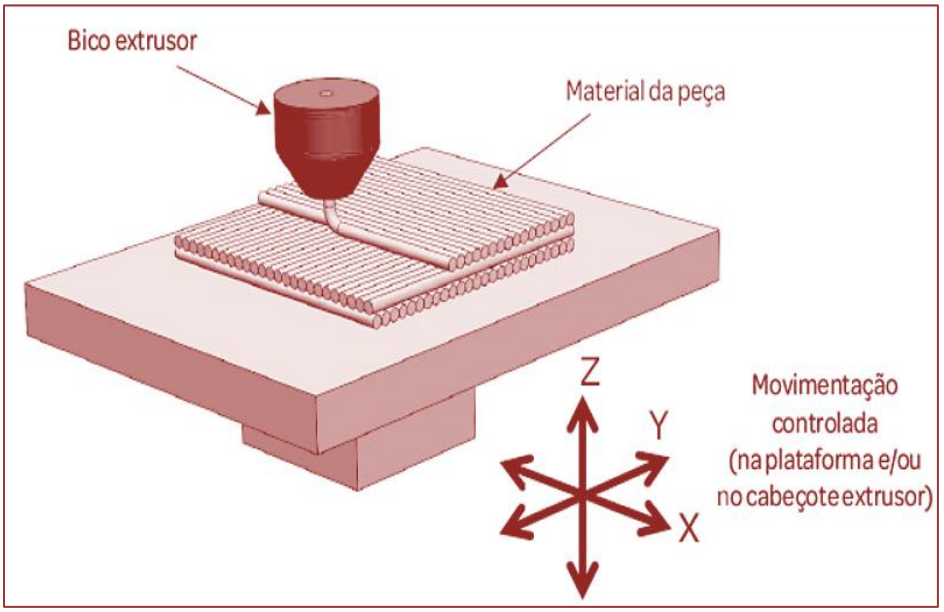
3.2.4. PROCESSO FDM (FUSED DEPOSITION MODELING)

Dentre as diversas tecnologias empregadas, atualmente, as pesquisas mostram o predomínio da aplicação da técnica Fused Deposition Modeling (FDM), em aplicações de pesquisa e desenvolvimento de produtos, com a aplicação de materiais como PLA, ABS e PETG (de Paula Netto ,2024).

O processo FDM constitui uma das tecnologias centrais da manufatura aditiva no âmbito da Indústria 4.0, conforme discutido por Beyca et al. (2018). Este método de fabricação aditiva baseia-se na extrusão precisa de materiais termoplásticos aquecidos, que são depositados camada por camada para formar objetos tridimensionais. Com uma resolução típica de camada em torno de 0,25 mm, o processo FDM destaca-se por sua capacidade de produzir geometrias complexas de maneira eficiente, sendo particularmente adequado para prototipagem rápida e produção de pequenos lotes.

Uma vez que a peça é produzida por deposição de material, camada por camada, conforme ilustrado na Figura 3.1, há pouco desperdício de materiais, o que torna o processo altamente sustentável. (Rodrigues, 2017).

Figura 3.1 - Representação do processo FDM



Fonte: Volpato (2021).

3.2.5. PARÂMETROS DE PROCESSO

Nesta seção são apresentados os principais parâmetros de processo durante impressão pelo processo FDM, estes parâmetros impactam diretamente na qualidade da peça produzida, resistência mecânica, tempos de fabricação e quantidade de material utilizado.

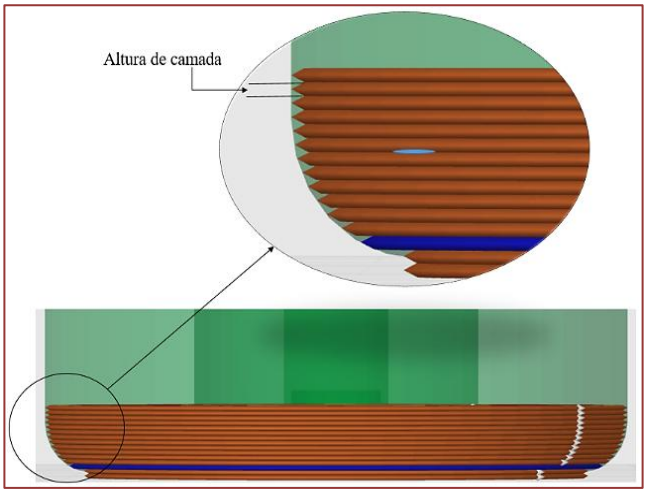
A otimização destes diferentes parâmetros é uma busca constante tanto no contexto acadêmico quanto industrial, uma vez que tem implicações diretas em características das peças produzidas, tais como adesão entre camadas, resistência mecânica, acabamento superficial, dentre outros. Manipular corretamente estes parâmetros permite aumentar o desempenho das peças produzidas ou ainda proporcionar um processo produtivo mais eficiente.

ALTURA DE CAMADA

A altura da camada é a espessura de cada camada que a impressora deposita durante o processo de impressão. É um dos principais parâmetros de processo, pois desempenha um papel significativo na aparência da peça, na precisão dimensional e na velocidade de impressão.

Este parâmetro é configurado no software de fatiamento, onde o projetista já consegue visualizar o efeito que a alteração deste item fará na peça produzida, conforme representado na Figura 3.2.

Figura 3.2 - Altura de camada no processo FDM

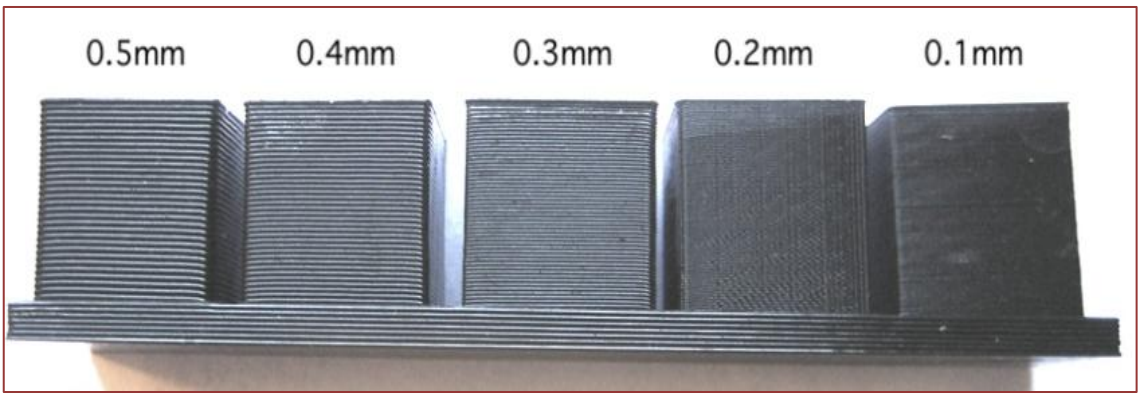


Fonte: Os autores, 2025.

Valores usuais para altura de camadas ficam em um range de 0,1 mm a 0,5 mm, dependendo da impressora e do bico extrusor utilizados. Um ajuste com valores altos de altura de camada é capaz de acelerar o processo de fabricação, uma vez que a impressora fabricará um menor número de camadas até chegar ao topo, contudo valores menores proporcionam um ajuste fino e suavidade na superfície de impressão, melhorando a

aparência da peça produzida.O efeito visual da altura de camada pode ser observado na Figura 3.3.

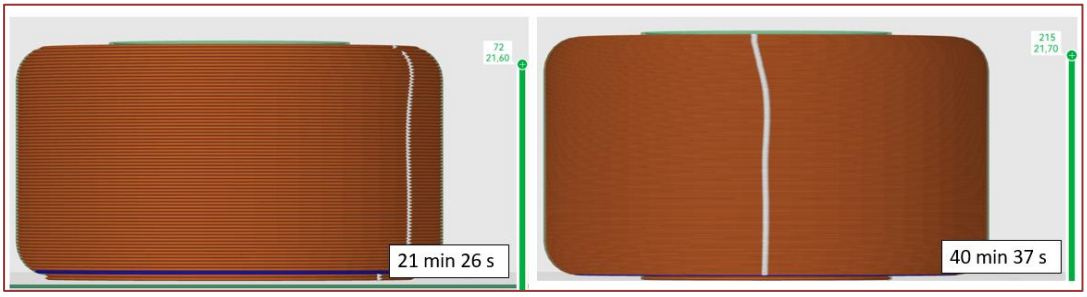
Figura 3.3: efeito visual de diferentes alturas de camadas



Fonte: Rodriguez (2024).

O aumento do tempo de processo ao utilizar uma altura de camada menor é explicado pela necessidade de fabricar um maior número de camadas de deposição de material até alcançar o topo da peça. A Figura 3.4 ilustra este comportamento, onde na figura (a) foi adotada uma configuração de altura de camada de 0,3 mm que resultou em 72 camadas para fabricar a peça, enquanto na figura (b) foi adotado uma altura de camada de 0,1 mm o que aumentou o número de camadas para 215. A alteração deste parâmetro resultou em um aumento de 89% do tempo para fabricação.

Figura 3.4 - Comparativo de tempo de processo: (a) Altura de camada de 0,3 mm; (b) altura de camada de 0,1 mm



Fonte: Os autores, 2025.

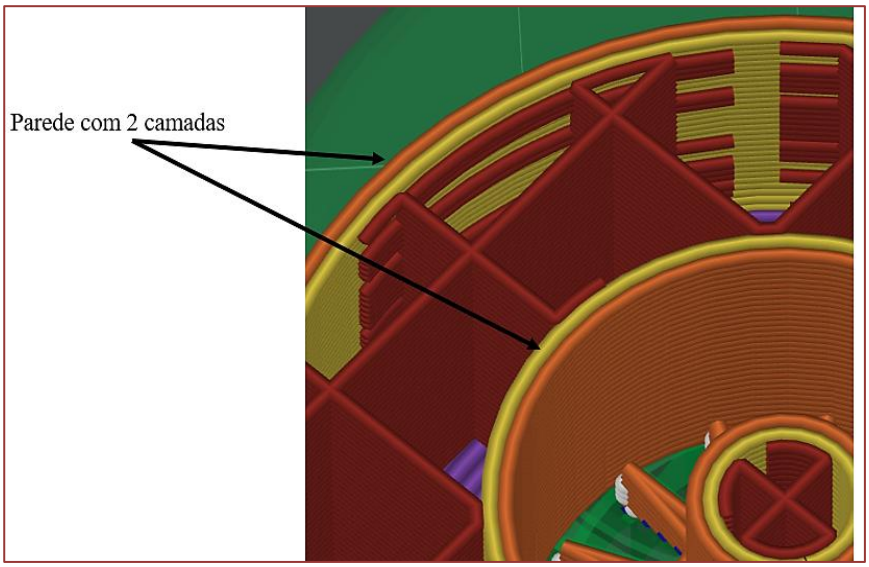
De forma geral, a escolha da altura de camada requer uma decisão entre melhor qualidade e acabamento superficial da peça ou priorizar a redução do tempo de fabricação. Portanto cabe um olhar crítico do projetista para entender que função a peça produzida irá executar e qual nível de acabamento superficial é suficiente para que a peça

produzida atenda a sua função sem tornar o processo produtivo demasiadamente oneroso.

LOOPS DE PAREDE

Este parâmetro determina a quantidade de camadas que cada parede de peça produzida possuirá, é um parâmetro importante quando fabricados protótipos funcionais ou peças que receberão esforço mecânico uma vez que, em geral, um maior número de camadas resultará em um aumento da rigidez mecânica da peça. Na Figura 3.5 o filamento na cor laranja representa a parede externa, enquanto o filamento na cor amarela representa a parede interna.

Figura 3.5 - Loops de parede



Fonte: Os autores, 2025.

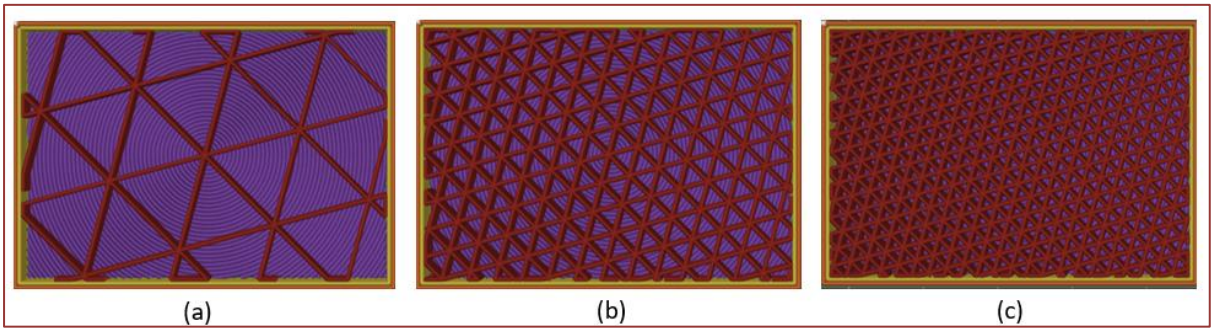
A altura da camada é a espessura de cada camada que a impressora deposita durante o processo de impressão.

PREENCHIMENTO INTERNO

O padrão de preenchimento é um dos parâmetros mais importantes na construção de peças por manufatura aditiva, a escolha adequada do padrão impacta diretamente na resistência mecânica, densidade e tempo de processo (Pereira, 2023).

Volpato (2021) explica que um preenchimento maior reduz o gap entre camadas de material, o que aumenta a resistência mecânica da peça fabricada (Figura 3.6). Já em casos em que a resistência mecânica não é tão relevante é desejável empregar uma deposição espaçada visando reduzir o tempo de fabricação e a quantidade de material utilizado.

Figura 3.6 - Adoção de diferentes percentuais de preenchimento: (a) 15%; (b) 50% (c) 80%

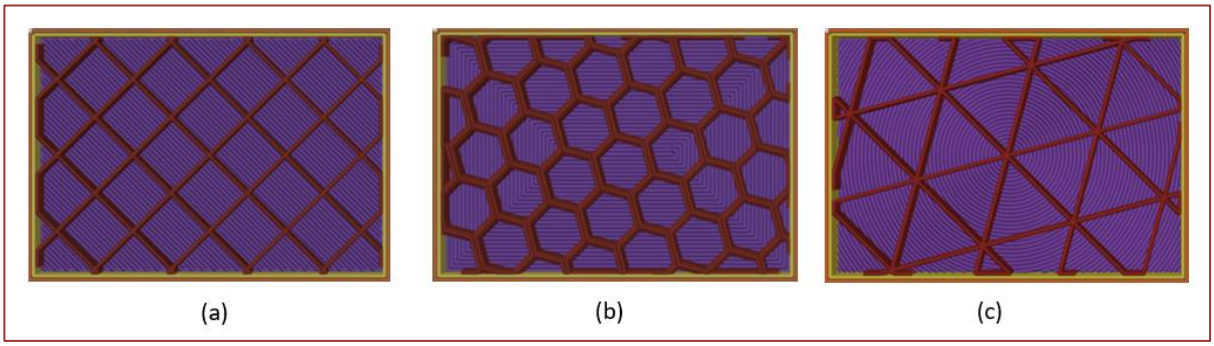


Fonte: Os autores, 2025.

ESTRATÉGIAS DE PREENCHIMENTO

A estratégia de preenchimento faz referência à trajetória adotada para realizar o preenchimento das camadas. Há uma grande variedade de estratégias e a nomenclatura varia de acordo com o software de planejamento de processo utilizado. Cada uma das estratégias possui um conjunto de parâmetros que controla o resultado do preenchimento; assim, é necessário recorrer a cada software de planejamento para a correta especificação desses parâmetros (Volpato, 2021). A Figura 3.7 apresenta três estratégias de preenchimento diferentes.

Figura 3.7 - Exemplos de preenchimento: (a) em grade; (b) colmeia (c) triangular



Fonte: Os autores, 2025.

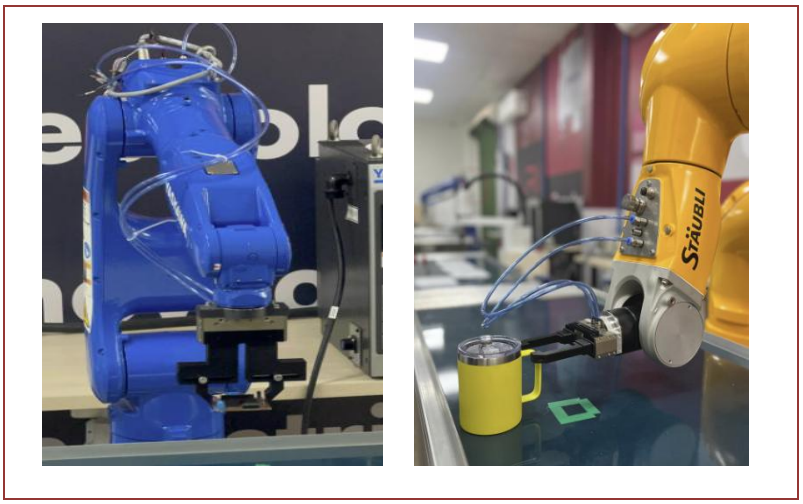
3.2.6. MANUFATURA ADITIVA NO CONTEXTO INDUSTRIAL

No setor industrial a manufatura aditiva tem ganhado relevância em atividades de desenvolvimento de soluções em robótica, principalmente pela capacidade de desenvolver componentes complexos com rapidez e flexibilidade (Volpato, 2021). Além da flexibilidade do processo e capacidade de fabricar peças com geometrias complexas, ao longo dos últimos anos o processo passou por constantes melhorias alcançando maior precisão e melhores propriedades mecânicas o que justifica uma crescente aplicação de peças fabricadas por este processo em um contexto industrial (Gibson, 2021).

A manufatura aditiva pode ser usada em uma variedade de aplicações de testes funcionais, no contexto da robótica podem ser desenvolvidas ferramentas para manipulação de peças ou desenvolvidos protótipos para validação de soluções definitivas por meio da fabricação de protótipos funcionais, peças de reposição e estruturas customizadas (Gibson, 2021).

A Figura 3.8 apresenta duas soluções diferentes de garras para manipulação de objetos que foram desenvolvidas no decorrer da disciplina. Os alunos foram desafiados a criar soluções para manipulação de componentes diversos, a fabricação das garras passou por todas as etapas do processo de manufatura aditiva, desde o projeto CAD, fatiamento em arquivo step e fabricação da peça por meio do processo FDM.

Figura 3.8 - Diferentes garras desenvolvidas no módulo de manufatura aditiva manipulando componentes de geometrias variadas



Fonte: Os autores, 2025.

Ao desenvolver ferramentas com geometrias otimizadas para determinada aplicação é possível reduzir o peso das ferramentas de extremidade que impactam diretamente na eficiência energética e na velocidade operacional dos sistemas robóticos.

3.2.7. CONSIDERAÇÕES FINAIS

A manufatura aditiva é um processo em constante evolução, que ganha cada vez mais relevância e se destaca pela acessibilidade, simplicidade e diversidade de materiais em comparação a processos de fabricação mecânica tradicionais.

Em um contexto industrial, caracterizado pela alta competitividade e pela necessidade de inovação constante, este processo permite a fabricação de peças e componentes complexos com rapidez, flexibilidade e custo relativamente reduzido. Atualmente o processo não se limita à prototipagem, mas também possibilita a fabricação de elementos funcionais.

No decorrer do módulo de Manufatura Aditiva os alunos puderam atestar a flexibilidade do processo ao projetar garras para manipulação de peças de diversos tamanhos e geometrias em diferentes robôs. Também foi possível estudar sobre vantagens e limitações do processo, como conclusão fica o entendimento de que o

conhecimento dos parâmetros de processo, associado à manutenção e operação adequada dos equipamentos, é essencial para explorar todo o potencial da tecnologia.

REFERÊNCIAS

[1] BEYCA, Omer Faruk; HANCERLIOGULLARI, Gulsah; YAZICI, Ibrahim. **Additive manufacturing technologies and applications. In: Industry 4.0: Managing The Digital Transformation**. Cham: Springer International Publishing, 2017. p. 217-234.

[2] DE PAULA NETTO, Miguel Olimpio et al. **A utilização da manufatura aditiva no desenvolvimento de novos produtos e tecnologias no brasil: uma revisão sistemática.** ARACÊ, v. 6, n. 4, p. 13674-13714, 2024.

[3] GIBSON, Ian et al. **Additive manufacturing technologies**. Cham, Switzerland: Springer, 2021.

[4] PEREIRA, Erisson Jeliel Mendes. **Avaliação do tipo de preenchimento na resistência mecânica de peças fabricadas em petg por impressão 3D**. 2023.

[5] Rodriguez Laurent. **Guia completo sobre a altura da camada na impressão**. Disponível em:<HTTPS://www.trideo3d.com/br/blog-posts/guia-completa-sobre-la-altura-de-capas-en-la-impresion-3d>. Acesso em: 01/08/202501 ago. 2025.

[6] RODRIGUES, Vinícius Picanço et al. **Manufatura aditiva: estado da arte e framework de aplicações. Revista Gestão da Produção Operações e Sistemas**, v. 12, n. 3, p. 1-1, 2017.

[7] USTUNDAG, Alp; CEVIKCAN, Emre. **Industry 4.0: managing the digital transformation.** Cham: Springer, 2018.

[8] VOLPATO, Neri. **Manufatura aditiva: tecnologias e aplicações da impressão 3D.** Editora Blucher, 2021.

CAPÍTULO

04 Introdução a robótica industrial

Benjamin Batista de Oliveira Neto[1], Ricardo Augusto Medeiros de Oliveira[2]

4. CARACTERIZAÇÃO DA DISCIPLINA

4.1. CARACTERIZAÇÃO DA DISCIPLINA

O curso de Robótica Industrial tem como objetivo geral promover e desenvolver estratégias de ensino e aprendizado, por meio da capacitação na área de robótica industrial, com enfoque em tecnologias emergentes conectadas às necessidades e potencialidades do Polo Industrial de Manaus.

Já a disciplina de introdução à Robótica Industrial objetiva proporcionar aos estudantes uma compreensão generalista dos conceitos fundamentais e aplicações dos sistemas robóticos industriais; apresentar princípios básicos da robótica, bem como aos diversos tipos de robôs utilizados na indústria; fornece uma visão geral das aplicações práticas dos robôs industriais em diferentes setores; identificar as partes que compõe os robôs industriais e realizar operações com os robôs industriais de manuseio e movimentação.

4.1.1. EMENTA

Introdução a robótica industrial: Definição; Breve Histórico; Aplicações e Benefícios. Aplicações Robóticas; Segurança em ambientes com robôs industriais; Conceitos e Tecnologias da Robótica Industrial: Quais são as partes de um robô; O manipulador mecânico; Punho do Robô; O efetuador final; Tipos de Acionamentos; Transmissões Mecânicas; Tipos de Algoritmos para Movimentação; Dinâmica do Braço Robótico: Precisão de movimentos; Velocidade de movimento; Estabilidade; Repetibilidade; Complacência, Cinemática Direta e Inversa; Servo acionamento Elétrico nos Robôs: Visão geral; Os sensores de movimento; Servomotores e seus drivers; Servomotor de corrente contínua; Drivers para servomotor DC; Motor de Passo; Drivers para motores de passo. Práticas de movimentação manual e automática nos robôs industriais (Yaskawa, Epson, Staubli e US-Robotics) através da programação em simuladores e nos equipamentos físicos; Noções de pneumática.

[1] Tecnólogo em Mecatrônica Industrial. Mestre em Ciência e Engenharia de Materiais. Professor da Disciplina de Introdução à Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM. E-mail: *benjamin.batista@ifam.edu.br*

[2] Engenheiro Mecânico. Mestre em Gestão e Estratégia, Professor da Disciplina de Introdução à Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM. E-mail: *ricardo.oliveira@ifam.edu.br*

4.1.2. OBJETIVOS

A disciplina de Introdução à Robótica Industrial tem como objetivos proporcionar aos estudantes uma compreensão generalista dos conceitos fundamentais da robótica industrial e de suas principais aplicações. Busca apresentar os princípios básicos da robótica, abordando os diversos tipos de robôs utilizados na indústria, suas características e funcionalidades. Visa também fornecer uma visão abrangente das aplicações práticas desses robôs em diferentes setores produtivos, além de possibilitar a identificação e o entendimento das partes que compõem um robô industrial, como manipuladores, atuadores, sensores e controladores. A disciplina se propõe ainda a desenvolver habilidades práticas de manuseio e movimentação manual dos robôs industriais disponíveis no Laboratório de Robótica Industrial do IFAM, bem como capacitar os alunos na execução de programas de movimentação manual e automática, utilizando os diferentes tipos de movimentos específicos de cada robô. Por fim, inclui a introdução aos princípios básicos da pneumática aplicada à robótica, com foco no estudo e reconhecimento dos principais componentes pneumáticos utilizados em sistemas automatizados.

4.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 30h

Carga Horária Prática: 30h

Total: 60h

A disciplina usou como atividade de avaliação prática a movimentação manual e a criação de um programa em cada um dos robôs estudados. (Epson, Yaskawa, Staubli e Us-Robotics); Apresentação de atividades práticas Hands On em grupo onde forma avaliado a participação, o interesse e a demonstração de conhecimentos.

Durante o decorrer da disciplina, foram trabalhados os seguintes conteúdos

- Introdução a robótica industrial: Definição; Breve Histórico; Aplicações e Benefícios. Aplicações Robóticas.
- Conceitos e Tecnologias da Robótica Industrial: Quais são as partes de um robô; O manipulador mecânico; Punho do Robô; O efetuador final; Tipos de Acionamentos; Transmissões Mecânicas; Tipos de Algoritmos para Movimentação.
- Dinâmica do Braço Robótico: Precisão de movimentos; Velocidade de movimento; Estabilidade; Repetibilidade; Complacência.
- Servoacionamento Elétrico nos Robôs: Visão geral; Os sensores de movimento; Servomotores e seus drivers; Servomotor de corrente contínua; Drivers para servomotor DC; Motor de Passo; Drivers para motores de passo.
- Programação Básica de Acionamentos, Movimentos e Rotinas nos Robôs Industriais.
- Noções de Pneumática.

4.2. CONTEÚDOS

4.2.1. ETAPA 1: INTRODUÇÃO A ROBÓTICA INDUSTRIAL: DEFINIÇÃO; BREVE HISTÓRICO; APLICAÇÕES E BENEFÍCIOS. APLICAÇÕES ROBÓTICAS

Esta etapa tem por objetivo apresentar as origens dos robôs industriais tais como os conhecemos na atualidade. Serão apresentadas algumas definições que esclarecem exatamente o que é um robô industrial, bem como uma visão geral sobre o campo de aplicações e benefícios em relação ao seu uso.

Apresenta ainda um histórico dos principais desenvolvimentos tecnológicos que marcaram os avanços da robótica, bem como o atual estado da técnica e as áreas de competência necessárias ao projeto e à fabricação de robôs.

Robô? Qual o significado dessa palavra?

Segundo o escritor Karel Capek (1890-1938): República Tcheca – Peças teatro: Rossum's Universal Robots, a sociedade buscava cada vez mais a materialização do conceito do que o significado da palavra robota (do idiona tcheco), naquele espetáculo de um "trabalhador escravo", feito por criaturas mecânicas de forma semelhante à humana.

Já o engenheiro elétrico Roy J. Wensley apresentou um primeiro mecanismo com aparência humana na Exposição Mundial de Nova York, em 1927 – Robô Televox (um pequeno robô de aspecto humano capaz de cumprir as ações ordenadas por seu operador).

Breve Histórico:

Anos 40: No final da década de 40 e início da de 50 são iniciadas as primeiras pesquisas com robôs reais.

1947: Impulsionados pelo programa da bomba atômica, o Laboratório Nacional em Argonne desenvolve o primeiro manipulador robótico tele-operado: – Manipulação de material radioativo

1968 – Shakey

O Stanford Research Institute demonstra o primeiro robô inteligente.

O robô é capaz de interpretar visualmente o meio onde está inserido, localizar objetos, navegar entre os objetos e raciocinar sobre suas ações (Santos e Júnior, 2014).

Foi nomeado "Shakey" por causa de seus movimentos erráticos e "Balançantes".

1968 A General Electric constrói o "Four Legged Walking Truck" para o exército americano: – Os movimentos são comandados por um operador dentro do robô, mecanicamente.

Cavaleiro mecânico de Leonardo

Robô de Leonardo (ou cavaleiro mecânico de Leonardo) foi um humanóide desenhado e possivelmente construído por Leonardo da Vinci por volta do ano 1495.

Todo o sistema robótico era operado por uma série de roldanas e cabos.

Desde a descoberta do caderno, o robô foi construído fielmente com base no design de Leonardo e descobriu-se que é totalmente funcional.

Para Mataric (2014) o robô é descrito como sendo coberto por uma armadura medieval alemã-italiana e é capaz de fazer vários movimentos parecidos com humanos.

Década de 1990.

Para Carrara (2014) o "baixo custo" e o aumento da capacidade dos microcomputadores possibilita a introdução dos robôs em novas atividades comerciais.

As aplicações de robôs se expandem para a indústria de serviços.

Início de pesquisa séria em robôs humanoides.

Início do uso doméstico. 1993

Honda começa a criar protótipos de robôs humanóides.

Modelo P1: – 195 cm – 175 kg.

Modelos P2 (1996) e P3 (1997) são protótipos para o Robô Asimo

Robótica Educacional: os robôs educativos mais populares

Existem atualmente numerosos robôs educativos para crianças e jovens. Entre os mais populares estão:

Makeblock mBot: este é um robô com rodas projetado para introduzir as crianças à robótica, à programação e à eletrônica. É fácil de montar e é facilmente controlado graças ao software baseado em Scratch, projetado para crianças. Sua compatibilidade com a plataforma Makeblock e suas peças eletrônicas baseadas no ecossistema de código aberto Arduino permitem aos usuários com mais conhecimento criar robôs mais complexos.

Robo Wunderkind: consiste em um conjunto de blocos que as crianças podem conectar como quiserem para construir seu próprio robô. Cada bloco tem uma função identificada com uma cor (câmera, microfone, sensores de movimento...) e, depois de construir seu robô, as crianças podem programá-lo através de um aplicativo para reagir a certos ruídos, evitar obstáculos ou tocar música quando alguém se aproxima, entre outras funções.

OWI 535: é um braço robótico recomendado para jovens a partir dos 13 anos de idade. Ele pode levantar objetos de até 100 gramas e tem uma grande variedade de movimentos que podem ser personalizados graças à programação. Este robô também é recomendado para cursos de treinamento vocacional.

LEGO Mindstorms Robot Inventor: o herdeiro do Lego Mindstorms EV3 Robot apareceu no final de 2020 e conta com 4 motores angulares, vários sensores e 900 peças que permitem a criação de pelo menos 5 tipos diferentes de robôs. Para controlá-lo, é necessário um tablet ou um computador para realizar sua programação. É recomendado para crianças com mais de 10 anos de idade

NAO: é um dos robôs educacionais mais conhecidos do mundo. Trata-se de robô humanoide de 58 centímetros de altura que está em constante evolução. Além de duas câmeras e quatro microfones, possui uma infinidade de sensores que lhe permitem interagir com o ambiente de uma maneira semelhante à humana. O NAO é capaz de observar, ouvir, manter diálogos e ensinar qualquer assunto. Graças a suas habilidades e diferentes níveis de programação, pode participar do processo de aprendizagem de estudantes de 5 anos de idade até a universidade,

LEGO Mindstorms Robot Inventor: o herdeiro do Lego Mindstorms EV3 Robot apareceu no final de 2020 e conta com 4 motores angulares, vários sensores e 900 peças que permitem a criação de pelo menos 5 tipos diferentes de robôs. Para controlá-lo, é necessário um tablet ou um computador para realizar sua programação. É recomendado para crianças com mais de 10 anos de idade.

Robótica Educacional

O interesse pela mobilidade de robôs teve início com hobistas das áreas de elétrica/eletrônica. A partir do final da década de 70 e início dos anos 80, começaram a aparecer projetos sérios para aplicações como patrulhamento de ambientes, transporte de carga, segurança, exploração, entre outros, quase sempre patrocinados por órgãos militares, governamentais ou grandes empresas (Barbosa e Blikstein, 2020).

A evolução nesta área é surpreendente. O caso mais famoso de projeto bem sucedido é o robô Sojourner. Construído pelo Jet Propulsion Laboratory (JPL) do Instituto de Tecnologia da Califórina em parceria com a NASA e enviado a Marte1 com o objetivo de explorar as características do terreno deste planeta e realizar experimentos científicos

AGV executa o mesmo trajeto de entrega ao longo de sua vida útil e para navegar precisa ser guiado por fios, tiras magnéticas ou sensores, que normalmente requerem atualizações das instalações.

Já, o AMR navega através de mapas “carregados” por seu software e também consegue se adaptar conforme a necessidade da instalação. Esta capacidade pode ser comparada a um carro com um GPS, por exemplo.

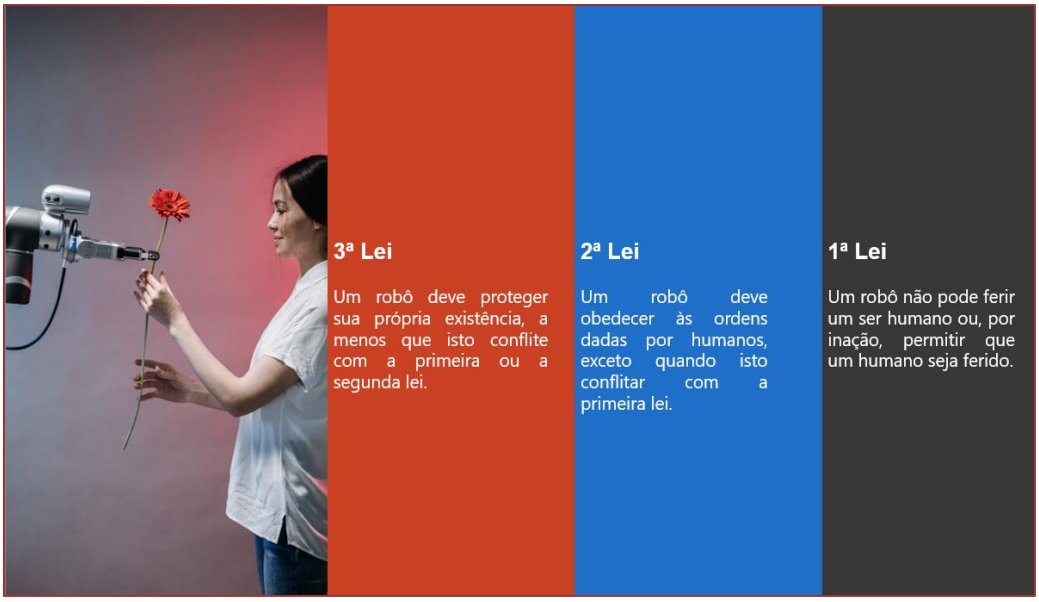
Dessa forma, ele proporciona uma navegação inteligente, que permite detectar seus arredores e escolher a rota mais eficiente até o seu destino.

O AMR funciona de forma completamente autônoma e consegue fazer manobras com segurança, caso obstáculos, pessoas ou empilhadeiras surgirem em sua frente.

Quais as leis da robótica?

O escritor russo-americano Isaac Asimov ditou as 3 (três) leis da robótica, as quais servem, inclusive, para os atuais robôs industriais. A Figura 4.1 explica cada umas das leis elaboradas pelo escritor.

Figura 4.1: Leis da Robótica



Fonte: Adaptado de Mataric, 2014.

Robô Industrial

Tais máquinas em quase nada se assemelham à aparência humana. Ao contrário, são formadas por articulações mecânicas capazes de realizar os movimentos da tarefa a que se destinam, não importando, para isso, a aparência que venham a ter.

Definições:

Um robô industrial é um manipulador reprogramável, multifuncional, projetado para mover materiais, peças, ferramentas ou dispositivos especiais em movimentos variáveis programadas para a realização de uma variedade de tarefas (Robot Institute of America – RIA)

A ISO 10218 define um robô industrial como uma máquina para manipulação, com vários graus de liberdade, controlada automaticamente, reprogramável, multifuncional, que pode ter base fixa ou móvel para utilização em aplicações de automação industrial. (*Organização Internacional de Normalização).*

Projeto e Engenharia

O atual estado da arte de robôs industriais implica preponderantemente tarefas de controle de posicionamento e movimentação, que é ainda o principal objetivo desses mecanismos.

Diversas capacidades de controle lógico de entrada e saída têm sido acrescentadas aos recentes robôs, assim como a possibilidade de comunicação.

Aplicação e benefícios

- Movimentação
- Medição
- Manipulação

4.2.2. ETAPA 2: CONCEITOS E TECNOLOGIAS DA ROBÓTICA INDUSTRIAL

O objetivo desta aula é descrever os principais fundamentos tecnológicos que compõem os sistemas robóticos atualmente empregados nas indústrias. O robô será descrito por cada uma de suas partes constituintes. Enfatizaremos o braço mecânico manipulador e seu efetuador final, assim como os principais tipos de garras.

Trataremos ainda das questões que envolvem o sistema controle de movimento do robô: tipos de tecnologias de acionamentos e sistemas de controle.

O robô está dividido em 6 (partes) como mostra a Figura 4.2.

Figura 4.2: Partes de um robô

Fonte: Os autores, 2023.

Manipulador Mecânico

Elos

São as partes rígidas do manipulador.

Juntas

São as partes do manipulador que realizam uma conexão móvel entre dois elos.

Efetuador Final

Localizado ao final da cadeia cinemática que compõe o manipulador é capaz de aplicar amplo grau de liberdade em termos de orientação espacial.

Juntas

Segundo Carraca (2014) as juntas de um robô são elementos de particular interesse na análise de um manipulador robótico. Afinal, são elas que proporcionam

liberdade de movimento. Dentre várias formas mecânicas construtivas possíveis de serem encontradas para juntas, há nos robôs industriais três tipos mais comuns.

Juntas Prismáticas

Também chamadas de juntas lineares ou deslizantes, elas se movem em linha reta sem girar.

São compostas de duas hastes que deslizam entre si de forma telescópica.

Elas se estendem, retraem-se ou se movem para dentro e para fora, como um elevador tipo macaco hidráulico, ou ainda ao longo de um trilho, como o cabeçote de uma impressora**.**

Juntas Rotacionais

Também chamadas juntas de revolução, giram em torno de uma linha imaginária estacionária, chamada de eixo de rotação.

Elas giram como em uma cadeira giratória, ou abrem e fecham como em uma dobradiça.

Juntas Esféricas

Esta conexão funciona como a combinação de três juntas de revolução, realizando a rotação em torno de três eixos.

No corpo humano existem algumas juntas esféricas, como ombro, punho e quadril.

Grau de Liberdade

O termo pode ser abreviado apenas para GDL ou, como frequentemente se encontra em catálogos de robôs, pela sigla equivalente em inglês, DOF (Degrees of Freedom).

Tipos de Manipuladores

Os robôs são normalmente classificados conforme o tipo de juntas, ou, mais exatamente, pela configuração das juntas que formam seu sistema de articulação. Esta divisão em classes possibilita categorizar certas características comuns, tais como:

Robô Cartesiano

Neste tipo de estrutura, os três elos deslocam-se de forma translacional, aqui denominada TTT.

Devido à ortogonalidade entre os elos mecânicos obtém-se boa rigidez mecânica do conjunto, porém há baixa habilidade de posicionamento devido à limitada orientação angular com que efetuam as tarefas

Robô Cilíndrico

A estrutura cilíndrica difere da estrutura cartesiana pelo fato de que a primeira junta mecânica provê movimento de rotação, em vez de translação, e por isso esta configuração é denominada RTT.

Esta estrutura ainda apresenta características de rigidez mecânica satisfatória, apesar de ter menor resolução de posicionamento na direção horizontal

Robô Esférico

A estrutura esférica é composta por dois elos mecânicos com movimento rotacional, seguidos por um terceiro, com movimento translacional, e por isso é denominada RRT.

Em função desta configuração, os movimentos apresentam menor resolução de posicionamento ao longo do eixo radial do robô, além de comportarem-se com menor grau de rigidez mecânica.

Robô SCARA

SCARA - Selective Compliance Assembly Robot Arm.

Nesta configuração, apresentam-se dois elos mecânicos com movimento rotacional, seguidos por um terceiro, com movimento translacional, e também é denominada RRT, como no caso do robô esférico.

Esta estrutura apresenta alta rigidez dinâmica para movimentos verticais. Ao mesmo tempo, para movimentos horizontais, a rigidez é reduzida.

Robô Articulado

Neste tipo de estrutura, os três elos mecânicos de posicionamento deslocam-se de forma rotativa - RRR.

Tal estrutura, apesar de ter menor rigidez mecânica do que as anteriores, assim como resolução de posicionamento variável no espaço de trabalho, é a que apresenta maior grau de habilidade para posicionamento e orientação espacial.

Robôs Colaborativos (Cobots)

Segundo a Associação Brasileira de Normas Técnicas (2018) um cobot ou robô colaborativo é um robô criado para interagir com humanos em ambientes de trabalho. O termo foi usado pela primeira vez em 1999 e resulta da junção das palavras "colaboração" e "robot". Quais são as suas principais características?

Colaboração: Os cobots foram criados para ajudar as pessoas, portanto a interação com os operários é fundamental.

Exige menos adequações físicas da área de trabalho para adequação as normas de segurança.

Normas complementares foram desenvolvidas para lidar especificamente com a robótica colaborativa. Como é o caso da TS 15066 e ISO 10218, que no Brasil foram incorporadas como ABNT NBR ISO 10218-1, ABNT NBR ISO 10218-2 e ISO/TS 15066.

Punho do Robô

Punho é o nome normalmente dado às três últimas juntas do braço do robô, o número de graus de liberdade para orientação depende do punho que em geral, é formado por juntas rotacionais (Santos e Júnior, 2015).

Idealmente seus eixos de rotação são mutuamente perpendiculares, configurando o que se conhece como punho esférico, ou seja, punhos cujos eixos das juntas se cruzam em um único ponto.

Uma das vantagens do punho esférico é que seu uso simplifica consideravelmente a cinemática do manipulador, permitindo o desacoplamento entre a posição e a orientação do efetuador.

O braço e o punho são utilizados basicamente para o posicionamento e a orientação do efetuador final, seja na forma de garra ou de qualquer ferramenta que ele possa carregar.

Efetuador Final

O último componente na sequência cinemática de um robô industrial é o seu efetuador, localizado imediatamente após o punho. Os efetuadores finais podem ser divididos em dois principais grupos: Garras e ferramentas.

Tipos de Acionamentos

Como você deve se recordar da Física, para alterar o movimento de um corpo é necessário aplicar uma força externa a ele. Tecnologicamente falando, para gerar força é necessário haver algum dispositivo motor. Em robótica industrial há basicamente três tecnologias principais para isto, ou seja, três tipos de fontes de força motriz para acionamentos, vide Figura 4.4.

Figura 4.4: Tipos de Acionamentos

Fonte: os autores, 2023.

Acionamento hidráulico:

Vantagens

- Apresenta maior disponibilidade de força/torque.
- Mantém esta força/torque constante para uma grande faixa de variação de velocidade.
- Possui média precisão de posicionamento (menor que o elétrico e maior que o pneumático).
- Pode manter uma elevada força (ou torque) por um longo período de tempo, mesmo quando parado

Desvantagens

- Requer uma fonte de energia mais cara.
- Depende de constante manutenção.
- Utiliza válvulas de precisão que ainda apresentam custo elevado.
- Está sujeito a vazamento de óleo e, portanto, é potencialmente poluente.

4.2.3. ETAPA 3: DINÂMICA DO BRAÇO ROBÓTICO

O objetivo desta etapa é proporcionar uma compreensão abrangente dos princípios fundamentais que regem o movimento e o controle de braços robóticos. Ao final da aula, os alunos devem estar preparados para compreender os princípios fundamentais da dinâmica de braços robóticos, bem como sua aplicação em situações do mundo real, e devem ser capazes de analisar e resolver problemas relacionados à dinâmica e controle de robôs.

Para Carraca (2014) a dinâmica do robô refere-se como ele realiza sua Locomoção, ou seja, à maneira como um corpo (no nosso caso, um robô) se desloca de um lugar para outro. O termo vem do latim *locus*, que significa "lugar", e *movere*, que significa "mover". Neste contexto, o desempenho dinâmico do braço robótico está associado à alguns fatores, tais como:

- Precisão
- Velocidade de resposta
- Estabilidade

Precisão do Movimento

Relaciona-se com a velocidade e estabilidade, pois é uma medida de erro na posição do órgão terminal.

A precisão de movimento está intrinsecamente correlacionada com três características dos braços:

Resolução Espacial

A resolução espacial é influenciada pelas características do sistema de controle e pelas imprecisões. O sistema de controle é encarregado de gerenciar todos os incrementos individuais das articulações. As imprecisões são determinadas pela qualidade dos componentes que compõem as conexões entre as articulações, tais como folgas nas engrenagens, tensões nas polias e histereses mecânicas e magnéticas, entre outros fatores.

Precisão

Refere-se à habilidade de um braço em posicionar seu pulso em um ponto específico dentro do volume de trabalho. Está intrinsecamente ligada à resolução espacial, já que a precisão depende dos incrementos que as articulações podem realizar para movimentar-se e alcançar um ponto determinado. A precisão é impactada por vários fatores, e é crucial definir as condições nas quais ela será avaliada: no final do punho; na pior condição (geralmente com o braço totalmente estendido); no modo ponto-a-ponto de controle.

Repetibilidade

A repetibilidade está relacionada com a capacidade do braço robótico de posicionar repetidamente seu pulso num ponto determinado.

Estes movimentos podem sofrer influências de folgas mecânicas, da flexibilidade e das limitações do sistema de controle**.**

Erro de Posicionamento

Devido a várias razões, o braço não para precisamente no ponto indicado, apresentando um erro. O ponto-alvo é, de fato, o ponto desejado; na pior situação, o ponto-alvo está localizado exatamente no meio de dois pontos endereçáveis.

Precisão do Movimento

A precisão de um braço também depende da sua geometria.

Robôs pórticos, ou cartesianos, possuem uma grade de endereçamento perfeitamente retangular, os pontos endereçáveis formam um padrão cúbico.

Nos braços que utilizam juntas rotativas a grade de pontos endereçáveis não é regular.

Estabilidade

A estabilidade está relacionada com a maneira pela qual o braço alcança sua posição final com velocidade zero. Se a estabilidade for baixa, é possível aplicar elementos de amortecimento no braço para melhorar a estabilidade, mas isso afeta a velocidade de resposta. A maioria dos robôs precisa ser estável, o que significa que, ao realizar suas tarefas, não devem balançar, inclinar-se ou cair facilmente.

Existem diferentes formas de alcançar a estabilidade, e em particular, destacam-se dois tipos: estática e dinâmica. Um robô estaticamente estável pode permanecer parado sem cair, sendo, portanto, estático e estável. Isso é um recurso útil, mas requer que o corpo do robô tenha pernas ou rodas suficientes para proporcionar pontos de apoio estáticos capazes de mantê-lo em equilíbrio.

Na estabilidade dinâmica, o corpo precisa equilibrar-se ativamente ou mover-se para permanecer estável, sendo assim chamado dinamicamente estável. Por exemplo, robôs que saltam com uma perna são dinamicamente estáveis: podem pular sem se deslocar ou saltar para diferentes destinos sem cair.

Velocidade

Refere-se à agilidade do braço robótico ao deslocar-se de um ponto a outro em um curto período de tempo. Durante o projeto do manipulador, é crucial analisar tanto o torque presente em cada junta do braço quanto a aceleração de cada elo.

A velocidade das juntas exerce uma forte influência na duração do ciclo de trabalho e na quantidade produzida. Por outro lado, características como precisão no posicionamento do braço, carga (peso) do objeto manipulado e alcance do braço impactam diretamente na velocidade.

Não há almoço grátis: para um mesmo tipo de acionamento, quanto maior a velocidade, menor será a precisão e a capacidade de manipular cargas pesadas.

Complacência

Complacência é a capacidade do braço robótico de resistir ao movimento causado por uma força ou torque na extremidade do punho (Santos e Júnior, 2014).

A complacência é direcional e pode ser diferente em direções distintas.

Um exemplo disto é o robô SCARA, que possui complacência seletiva nas juntas, isto é, a rigidez apresentada pelas juntas é maior na direção que suporta a carga, o que o torna mais preciso no posicionamento.

Em contraste, um braço com alta complacência pode deformar-se ao suportar uma carga, comprometendo assim a precisão.

Cinemática de manipuladores

Sistema de Coordenadas

Qualquer ponto no espaço pode ser visto de diferentes formas consoante o referencial usado; trata-se sempre do mesmo ponto fisicamente, mas é a sua descrição que se adequa com o ponto vista (referencial)

Transformação geométrica de translação

Movimentar um ponto no espaço significa alterar as coordenadas, isto é, passar de uma posição q para uma posição q visto do mesmo referencial; este processo designa-se por translação no espaço

Transformação geométrica de rotação

Mudança na posição de um ponto a partir de um movimento de rotação ao redor de um eixo.

A figura 3.5 mostra alguns momentos de atividades práticas para compreender os conteúdos abordados e esclarecimentos.

Cinemática Direta

A cinemática de um manipulador: é o estudo do conjunto de relações entre as posições, velocidades e acelerações dos seus elos (Carrara, 2014).

A relação entre o referencial de origem e o referencial da extremidade (mão) do manipulador é dada por uma transformação T.

Para fazer um estudo da cinemática, e outros, é necessário, antes de mais, definir sistemas de coordenadas associados a cada elo, ou seja, atribuir-lhe referenciais

A cinemática direta: consiste em achar a posição do manipulador a partir da posição das juntas do braço robótico.

A cinemática inversa: consiste em achar a das juntas do braço robótico a partir do manipulador.

Singularidade

É uma situação (configuração do robô) na qual seria necessário impor velocidades infinitamente altas numa ou mais juntas para manter determinadas velocidades no espaço operacional.

Haverá um atuador (ou mais) cujo controle será sempre insuficiente para cumprir as velocidades desejadas no espaço operacional.

Numa situação de singularidade há a perda de um ou mais graus de liberdade.

4.2.4. ETAPA 4: SERVOACIONAMENTO ELÉTRICO NOS ROBÔS:

O objetivo desta etapa é apresentar os elementos que compõem um sistema elétrico de servoacionamentos empregados nos robôs. O tema principal da aula será o controle de movimento, tratando mais especificamente dos servossistemas de posicionamento. Será analisado aspectos construtivos, métodos de operação e drivers para acionamento.

Para Silva e Blikstein (2014) o servossistema é um conceito destinado à realização de controle de qualquer grandeza física, e pode ser classificado como sendo de malha aberta ou de malha fechada.

Um sistema de malha aberta é aquele em que a saída do sistema não é diretamente controlada com base no feedback. Em vez disso, a entrada é definida para um valor desejado, e o sistema tenta produzir a saída desejada sem monitorar ou corrigir erros.

Um sistema de malha fechada é aquele em que a saída do sistema é continuamente monitorada e comparada com uma referência (valor desejado). Qualquer diferença (erro) entre a saída real e desejada é usada para ajustar a entrada e, assim, corrigir o sistema.

Em se tratando de robôs, as grandezas físicas de interesse são movimentos, e por isso é importante que o profissional da área conheça os componentes e os princípios tecnológicos que regem os servossistemas de acionamentos.

Atualmente, há uma tendência tecnológica cada vez maior ao uso de motores do tipo elétrico.

Para sistemas de posicionamento, o motor elétrico usado em um robô deve ser: motor de corrente contínua, motor de passo, ou motor síncrono.

Como um robô possui diversos elos móveis, o sistema completo de controle de movimento nada mais é do que o conjunto de diversos subsistemas de servoacionamentos.

Sensores de Movimento

Qualquer tipo de servossistema opera com base no conceito de realimentação negativa.

No caso dos robôs industriais, o interesse fundamental é o de controlar a posição do braço.

Em princípio, existe uma grande variedade de dispositivos disponíveis para detectar posição angular ou linear.

Potenciômetros

Os potenciômetros são dispositivos analógicos cuja tensão de saída é diretamente proporcional à posição do cursor. Ao contrário dos potenciômetros convencionais utilizados em circuitos eletrônicos, muitas vezes esses dispositivos não têm restrições no movimento circular, permitindo assim realizar múltiplas voltas.

Resolvers

O resolver é outra categoria de dispositivo analógico, cuja saída varia proporcionalmente ao ângulo de um elemento rotativo em relação a um elemento fixo. Na sua configuração mais básica, um resolver possui um único enrolamento em seu rotor e um par de enrolamentos no estator.

Encorder

O encoder é um dispositivo que registra e gera movimentos rotacionais. Esses dispositivos são empregados para medir tanto movimentos rotativos quanto deslocamentos lineares, convertendo-os em impulsos elétricos, sendo assim classificados como geradores de pulsos. Existem dois tipos principais de encoders: os incrementais e os absolutos.

Encoder incremental

O *encoder* incremental fornece o número de pulsos a partir de cada rotação inteira do eixo. Ou seja, a medida da distância angular ou linear movimentada é representada por essa quantidade. Cada pulso surge por meio do trabalho de um emissor luminoso e um receptor.

Equipamento que possui este tipo de encoder necessita referenciamento toda vez que que energizado.

Encoders absolutos

Os *encoders* absolutos fornecem informações sobre a posição real, permitindo identificar a posição exata mesmo após uma interrupção de energia, sem a necessidade de reinicialização. Essa característica é especialmente útil em aplicações onde a precisão e a continuidade são essenciais, como sistemas de controle, robótica e máquinas CNC.

Tacômetro

Um dos dispositivos mais comumente usados para a realimentação de informações de velocidade é o *tacômetro* de corrente contínua. Um tacômetro é essencialmente um gerador de corrente contínua, que fornece uma tensão de saída proporcional à velocidade angular do induzido

Servoacionamentos e seus Drivers

Os movimentos executados pelo motor são definidos pelo componente chamado controle, que estabelece parâmetros como torque, velocidade, distância, direção ou aceleração. Independentemente do tipo de motor utilizado (passo, DC ou síncrono), a função de controle é sempre existirá.

Servomotor de Corrente Contínua

Os motores DC foram a primeira alternativa adotada pelos projetistas de sistemas robóticos em substituição aos acionamentos hidráulicos.

Diferentes tipos construtivos de motores de corrente contínua são atualmente conhecidos, e os mais comuns são aqueles denominados núcleo de ferro (Iron Cored), Bobina Móvel (Moving Coil), e Sem Escovas (Brushless).

Como esses tipos de motores não apresentam núcleo de ferro girante imerso em campo magnético, também não apresentam um fenômeno denominado perdas no ferro.

Consequentemente, maiores velocidades podem ser obtidas com baixa potência de entrada.

O maior fator limitante no desempenho dos motores de núcleo de ferro ou bobina móvel é o excessivo calor gerado no rotor devido à corrente que circula nos enrolamentos.

Atualmente, esta categoria construtiva de motor DC é a mais difundida para uso em sistemas robóticos. Para se obter maiores torques com menores inércias, os fabricantes normalmente utilizam ímãs permanentes feitos com um material chamado terras-raras.

Por fim, indiferente ao tipo construtivo, outro aspecto importante nos servomotores DC diz respeito à sua especificação.

Drivers de para servomotor DC

O driver, também conhecido como amplificador de potência, em um sistema de servoacionamento elétrico, é o componente responsável por fornecer ao servomotor a energia elétrica necessária (tensão, corrente) para que ele se mova de acordo com os sinais de comando recebidos na sua interface de entrada.

Em geral, os servoacionamentos são do tipo reversível, ou, em uma expressão tecnicamente mais precisa: para funcionamento em quatro quadrantes. Isso significa que o driver pode fornecer energia tanto para acionar o motor (em ambos os sentidos de giro) como para receber energia durante a frenagem do mesmo (igualmente em ambas as direções de movimento).

Três métodos básicos são utilizados para controlar a tensão em amplificadores chaveados: modulação por largura de pulso (PWM – pulse width modulation), modulação da frequência de pulso (PFM – pulse frequency modulation), utilizando transistores, e conformação de onda (WS – wave shaping), utilizando tiristores (SCR – silicon controlled rectifier).

Para entender cada um desses métodos, é importante ter em mente que um motor DC responde mecanicamente da seguinte maneira: sua velocidade está diretamente relacionada à tensão elétrica a ele aplicada, enquanto o torque (ou “força” no eixo) está diretamente relacionado à corrente elétrica consumida pelo motor.

O motor de passo

Para Carrara (2014) esses motores transformam pulsos elétricos em movimentos mecânicos de incremento fixo.

Informa ainda que devido à natureza digital, são eficientes para operar em malha aberta, ou seja, não é necessário utilizar elemento sensor no eixo de saída.

Segundo Santos e Júnior (2014) os motores de passo apresentam boa estabilidade quando parados, não apresentando erros acumulativos.

Geralmente têm baixa necessidade de manutenção e suportam sobrecargas mecânicas sem causar prejuízos ao motor.

Talvez a principal vantagem advinda da aplicação com motores de passo seja o menor custo, pois se uma aplicação pode ser realizada com motor de passo, então esta deve ser a escolha mais adequada.

Os motores de passo também apresentam estabilidade em paradas, pois sem a comutação na corrente de fase o rotor permanece completamente parado, e o rotor não tende a vibrar em torno de uma posição do encoder ou do resolver (como pode ocorrer nos demais tipos de servomotores).

Fornecem altos torques de saída em baixas velocidades.

A perda de suavidade em baixas velocidades é o maior problema. Já para altas velocidades (~2000 rpm) começam a surgir ressonâncias mecânicas, o que os torna inadequados para operação em determinada faixa de velocidade.

O motor síncrono (*Brushless AC*)

Sem sombra de dúvidas, a tecnologia de ponta atual em termos de servoacionamento elétrico é definida pelo uso dos motores síncronos.

Popularmente, este tipo de motor é chamado de brushless (do inglês, brush quer dizer escova; portanto, brushless significa sem escovas).

Motores síncronos brushless (ou BLDC - Brushless Direct Current) são uma variante dos motores síncronos AC que não possuem escovas mecânicas. Em vez disso, eles usam ímãs permanentes e eletrônica de controle para gerar um campo magnético rotativo. Isso resulta em menos desgaste, maior eficiência e vida útil mais longa em comparação com os motores síncronos tradicionais.

Esses motores apresentam razão potência/volume superior à de servomotores de corrente contínua.

Drivers de para servomotor *Brushless AC*

O driver para um motor brushless utiliza a tecnologia de inversores, que são circuitos capazes de converter corrente contínua em corrente alternada, de maneira semelhante aos sistemas empregados em motores síncronos CA e motores de indução convencionais.

O princípio fundamental de operação de um inversor reside no controle da amplitude e da frequência da tensão na saída, por meio do chaveamento dos transistores que compõem o driver.

4.2.5. ETAPA 5: PROGRAMAÇÃO BÁSICA DE ACIONAMENTOS, MOVIMENTOS E ROTINAS NOS ROBÔS INDUSTRIAIS

Esta etapa do treinamento deve ser desenvolvida paralelamente às atividades teóricas e consiste em fazer as atividades práticas de movimentação e programação nos robôs dos diferentes fabricantes, Yaskawa, Us-Robotics, Epson e Staubli. Durante esta etapa os devem ter oportunidade de utilizar os robôs na prática, desde as movimentações mais simples, até a elaboração de programas de movimentação e simulação básica dos movimentos necessários para cumprir atividades práticas previamente definidas. A Metodologia utilizada nesta etapa deve ser a realização de atividades de forma individual e em equipes com a utilização dos próprios equipamentos bem como dos simuladores específicos de cada fabricante disponíveis no laboratório de robótica industrial. (Figuras 4.5 e 4.6).

Figura 4.5: Atividade Prática de Programação no Epson Scara e US-Robotics

Fonte: os autores, 2025.

Figura 4.6: Atividade Prática de Programação no Epson, Staubli e Yashikawa

Fonte: os autores, 2025.

4.2.6. ETAPA 6: NOÇÕES DE ELETROPNEUMÁTICA

Esta etapa do treinamento deve ser desenvolvida após as práticas com os robôs industriais e consiste em apresentar os fundamentos da pneumática aplicada à automação industrial. Durante esta etapa, os alunos devem ser introduzidos aos conceitos físicos do ar comprimido, tais como compressibilidade, elasticidade, expansibilidade e difusibilidade, além dos tipos de pressão e suas aplicações. Os estudantes terão contato com a estrutura dos sistemas pneumáticos, compreendendo os processos de produção, tratamento e distribuição do ar comprimido. Também serão apresentados os componentes principais como compressores, reservatórios, válvulas e atuadores, com foco em seu funcionamento e simbologia.

A metodologia a ser utilizada deve incluir atividades teóricas e práticas no laboratório, envolvendo o uso de sistemas reais de ar comprimido e componentes de controle, com o objetivo de proporcionar uma compreensão aplicada e integrada entre a robótica e a eletropneumática. A Figura 4.7 exemplifica uma atividade prática nesta área.

Figura 4.7: Atividade Prática de Eletropneumática

Fonte: os autores, 2025.

REFERÊNCIAS

[1] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISSO 10128-1: Robôs e dispositivos robóticos - Requisitos de segurança para robôs industriais Parte 1: Robôs.** Rio de Janeiro: ABNT, 2018.

[2] _____. **ABNT NBR ISSO 10128-1: Robôs e dispositivos robóticos - Requisitos de segurança para robôs industriais Parte 1: Sistemas robotizados e integração.** Rio de Janeiro: ABNT, 2018.

[3] Carrara, V. **Apostila de Robótica.** Mogi das Cruzes: Universidade Braz Cubas, 2014.

[4] Mataric, Maja J. **Introdução á robótica.** Editora Blucher, 2014.

[5] Santos, Winderson Eugênio dos.; Júnior, José Hamilton Chaves. **Robótica Industrial - Fundamentos, tecnologias, programação e simulação**. 1ª ed. Érica, 2015.

[6] Silva, Rodrigo B.; Blikstein, Paulo. **Robótica educacional:** experiências inovadoras na educação brasileira. Penso, 2019.

CAPÍTULO

05 Disciplina de programação para aplicações robóticas

Lenildo Marcos da Mota Silva[1], Ricardo Augusto Medeiros de Oliveira[2]

5.1. CARACTERIZAÇÃO DA DISCIPLINA

O curso de Robótica Industrial tem como objetivo geral promover e desenvolver estratégias de ensino e aprendizado, por meio da capacitação na área de robótica industrial, com enfoque em tecnologias emergentes conectadas às necessidades e potencialidades do Polo Industrial de Manaus.

5.1.1. EMENTA

A disciplina usou como atividade de avaliação prática a criação de programas em cada um dos robôs estudados. (Epson, Yaskawa, Us-Robotics e Staubli), bem como a apresentação de atividades práticas *Hands On* em grupo onde forma avaliado a participação, o interesse e a demonstração de conhecimentos. Durante o decorrer da disciplina, foram trabalhados os seguintes conteúdos:

- Partes do robô Controlador, Manipulador e interface de Programação.
- Coordenadas de Movimentação Manual.
- Coordenadas de usuário. (User Frames).
- Calibração de Ferramenta.
- Tipos de movimentos a serem programados.
- Instruções de controle (CALL,JUMP,LABEL, FOR etc.).
- Instruções de I/O entradas e saídas. Variáveis de controle.
- Identificação Binária de I/O. Instruções aritméticas.
- Funções de deslocamento. Criação de programas.
- Exemplos de aplicações.
- Programação Hand Free. Aplicações com Manipulação.
- Integração de Sistemas de I/O entradas e saídas.
- Programação Aplicada com robôs.

[1] Engenheiro de Automação e Elétrica Industrial. Professor da Disciplina de Programação para Práticas Robóticas. E-mail: *lenildo.marcos@gmail.com*

[2] Engenheiro Mecânico. Mestre em Gestão e Estratégia, Professor da Disciplina de Introdução à Robótica Industrial. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM. E-mail: *ricardo.oliveira@ifam.edu.br*

5.1.2. OBJETIVOS

A disciplina de Programação para aplicações Robóticas objetiva:

- Apresentar os princípios Funcionamento dos Módulos de programação dos robôs industriais abordados durante o curso;
- Proporcionar aos estudantes uma capacitação em programação e aplicações dos sistemas industriais;
- Desenvolver programas nas plataformas dos fabricantes utilizados durante o curso resolvendo possíveis problemas de programação que possam vir a existir em seus projetos

5.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 40h

Carga Horária Prática: 20h

Total: 60h

5.2. CONTEÚDOS

5.2.1. ROBÔ MANIPULADOR YASKAWA

O Robô Manipulador Yaskawa, também chamado de braço robótico, é um robô industrial muito utilizado no Polo Industrial de Manaus, e apresenta soluções modernas e totalmente aderente aos padrões propostos para a indústria 4.0. Devido à sua grande utilização na região, este foi um dos escolhidos para dar as bases conceituais e práticas da programação de robôs industriais. Os outros são o Robô Manipulador Universal COBOT, O Robô Epson SCARA e o Robô Manipulador Stäubli.

5.2.1.1. PARTES DO ROBÔ YASKAWA

O Robô Yaskawa é formado por controlador/unidade de potência, manipulador (com os sensores internos e externos) e interface de programação. O controlador é unidade central de controle do robô, e contém os modelos de software do manipulador e ambiente de programação, assim como o algoritmo de controlo do robô, que usa as informações dos sensores e do programa para comandar a unidade de potência.

O manipulador é estrutura mecânica constituída por diversos corpos rígidos (elos) ligados em cadeia, movimentados através dos atuadores que se encontram ligados aos mesmos através das respetivas transmissões. Neste caso, considerando o número de articulações e rotores, ele é um manipulador com 5 graus de liberdade. A Figura 5.1 apresenta a controlador/unidade de potência e o manipulador.

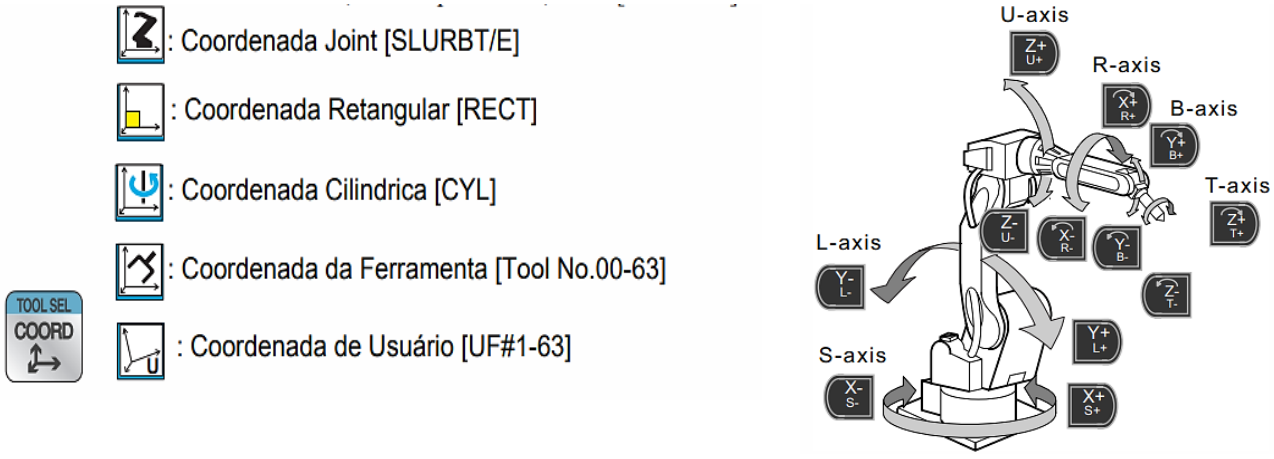
Figura 5.1 - Robô Yaskawa: Controlador (esquerda) e manipulador (direita)

Fonte: https://catalogo.yaskawa.com.br/, 2023.

5.2.2 SISTEMA DE COORDENADAS

O sistema de coordenadas é aquele que determina como o robô se movimentará, quando fica ativo se encontra identificado na área de status, e atua ao pressionar as teclas de movimentação. Quatro dos cinco tipos disponíveis estão visíveis na tecla COORD (há uma configuração disponível no menu TEACHING CONDITION para variar entre a coordenada RECT [Retangular] e CYL [Cilíndrica]). As teclas X, Y, Z, Rx, Ry e Rz ficarão ativas em todos os sistemas de coordenadas, exceto para Joint, onde [SLURBT] estarão ativas. A Figura 5.2 ilustra as teclas e as rotações em cada junta do robô.

Figura 5.2 - Teclas e as rotações em XYZ do sistema de coordenadas



Fonte: https://catalogo.yaskawa.com.br/, 2023.

Estas coordenadas são usadas para movimentação Manual do robô não sendo ainda inseridas no programa. A Coordenada Joint Robô de 6 eixos [SLURBT], quando selecionada, as letras S, L, U, R, B e T nos 06 pares de teclas dos eixos ficam ativas,

permitindo o operador mover cada motor do robô individualmente para qualquer direção. Essa é a coordenada ativa padrão quando o sistema é iniciado.

A coordenada TOOL movimenta o Ponto Central da Ferramenta (TCP), que fica "na ponta" do robô manipulador, baseando-se na *TOOL FRAME* (TF). Esse sistema de coordenada não é estacionário, ou seja, ele se movimenta junto com o robô. As teclas Z normalmente movimentam o robô para aproximação ou afastamento do ponto desejado, mas isso pode variar com a aplicação.

A coordenada de usuário (USER) movimenta o TCP baseando-se na coordenada ativa, onde os pontos e planos de origem são criados pelo usuário. A coordenada RECT movimenta o TCP em uma linha reta baseada no ROBOT FRAME (coordenada do robô), onde os eixos XYZ estão dispostos obedecendo a regra da mão direita, com o X movimentando o TCP para frente o para trás, o Y para a esquerda ou para a direita e o Z para cima e para baixo. A coordenada CYL movimenta o TCP em um movimento cilíndrico quando usamos as teclas XY; as teclas Z funcionam da mesma maneira.

5.2.3. PLANEJAMENTO DA PROGRAMAÇÃO

O planejamento da programação é uma etapa fundamental no desenvolvimento de aplicações com robôs industriais, pois garante que o processo seja estruturado, seguro e eficiente. Antes de iniciar a escrita do código no *Teach Pendant*, é necessário compreender com clareza a sequência de movimentos, a lógica operacional e os recursos que serão empregados na execução da tarefa. Esse planejamento reduz falhas durante a execução, evita colisões e otimiza o tempo de ciclo, além de assegurar maior precisão e repetibilidade do processo.

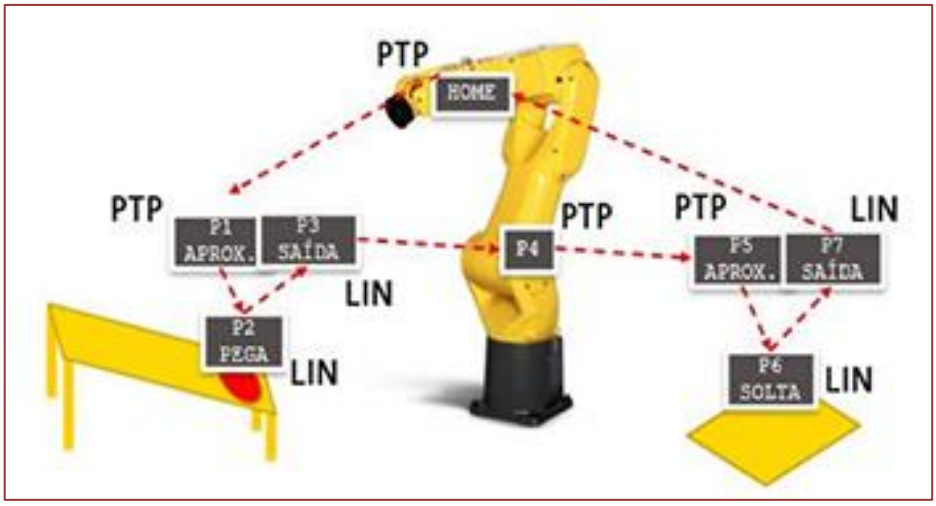
5.2.3.1. FERRAMENTAS MANUAIS

As ferramentas manuais desempenham um papel essencial no planejamento da programação e na construção do raciocínio lógico. Com o seu uso, o processo de programação torna-se mais dinâmico, estruturado e eficiente. Entre as principais ferramentas manuais utilizadas, destacam-se o Mapa de Pontos e o Fluxograma.

- Mapa de Pontos

O Mapa de Pontos é uma ferramenta utilizada para planejar e organizar a trajetória que o robô deverá executar em uma determinada tarefa. Ele representa graficamente os pontos de aproximação, de trabalho e de saída, permitindo visualizar de forma clara a sequência lógica dos movimentos. Na Figura 5.3, observa-se um exemplo de mapa de pontos aplicado a uma programação de manipulação, em que o robô retira uma peça de uma rampa e a deposita em um palete, percorrendo posições previamente definidas para garantir segurança e precisão na operação.

Figura 5.3 - Mapa de pontos

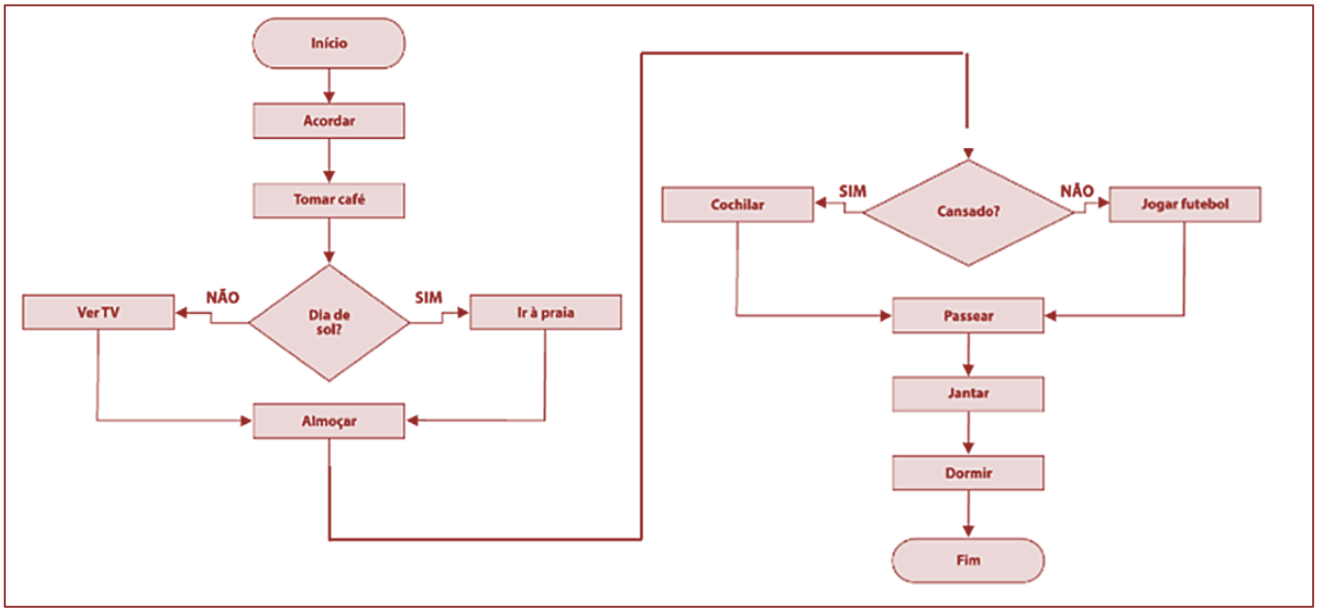


Fonte: https://catalogo.yaskawa.com.br/, 2023.

– Fluxograma

O Fluxograma é uma ferramenta gráfica utilizada para representar, de forma sequencial e organizada, as etapas de um processo. Ele faz uso de símbolos padronizados para indicar ações, decisões e pontos de início ou término, facilitando a compreensão da lógica de execução. Por meio dessa representação, é possível identificar falhas, sobreposições ou melhorias no processo antes da programação definitiva. Nas Figura 5.4, observa-se exemplo prático de sua aplicação em um processo simples, evidenciando a clareza e objetividade que essa ferramenta proporciona.

Figura 5.4 - Exemplo de um fluxograma



Fonte: https://catalogo.yaskawa.com.br/, 2023.

5.2.4. INTERFACE DE PROGRAMAÇÃO

O programa é feito através de uma de uma interface homem-máquina (IHM) chamada de programador pendente (P.P), mais conhecido pelo termo em inglês *Teach Pendant* (TP), que é o dispositivo mais comum na programação de robôs industriais, bastante utilizado para programar e controlar os movimentos dos robôs industriais e seus braços mecânicos passo a passo. Neste, são gravados os dados de posicionamento, velocidade e cálculos com variáveis dos movimentos feitos via o TP (Figura 5.5).

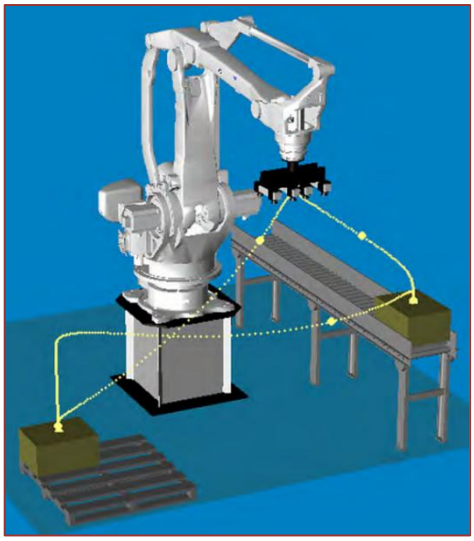
Figura 5.5 - *Teach Pendant* do Robô Yaskawa

Fonte: https://catalogo.yaskawa.com.br/, 2023.

5.2.5. TIPOS DE MOVIMENTOS

O *Joint Motion Type* (MOVJ) é o movimento que controla apenas o destino do robô. O *Control Group* se movimenta até alcançar os valores de encoder gravados na posição. A trajetória do robô não pode ser controlada, pois o controlador não interpola sua trajetória. Por isso, aplicações que precisem de precisão e trajetória conhecida (solda, pintura, etc.) não utilizam movimentos em Joint. Essa movimentação só é utilizada para movimentos aéreos e de aproximação. A velocidade *VJ=Joint_speed* representa a velocidade máxima de cada motor do robô durante a movimentação para a posição gravada. (Em alguns movimentos alguns motores podem não se mover, de acordo com a necessidade). A Figura 5.6 ilustra este tipo de movimento. (Yaskawa, 2014).

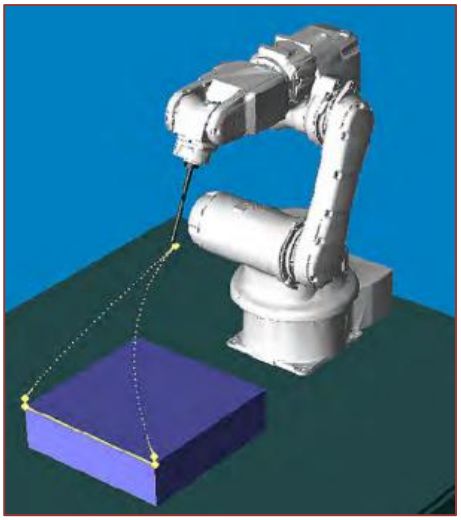
Figura 5.6 - Ilustração do movimento *Joint Motion Type* (MOVJ)



Fonte: https://catalogo.yaskawa.com.br/, 2023.

O *Linear Motion Type* (MOVL) é um movimento linear interpolado do TCP. Esse tipo de movimentação faz com que o TCP do robô se movimente em uma linha reta até chegar à posição gravada. Se houver uma mudança de orientação da ferramenta entre a posição atual para o destino, esta irá mudar gradativamente durante a trajetória linear até o destino. A velocidade do movimento MOVL é a velocidade com que o TCP se move (V= *Control_point_speed*) ou a velocidade de rotação (ou velocidade angular) da ferramenta (*VR=Angle_speed*), quando o robô rotaciona a ferramenta com o TCP fixo. A Figura 5.7 ilustra este tipo de movimento.

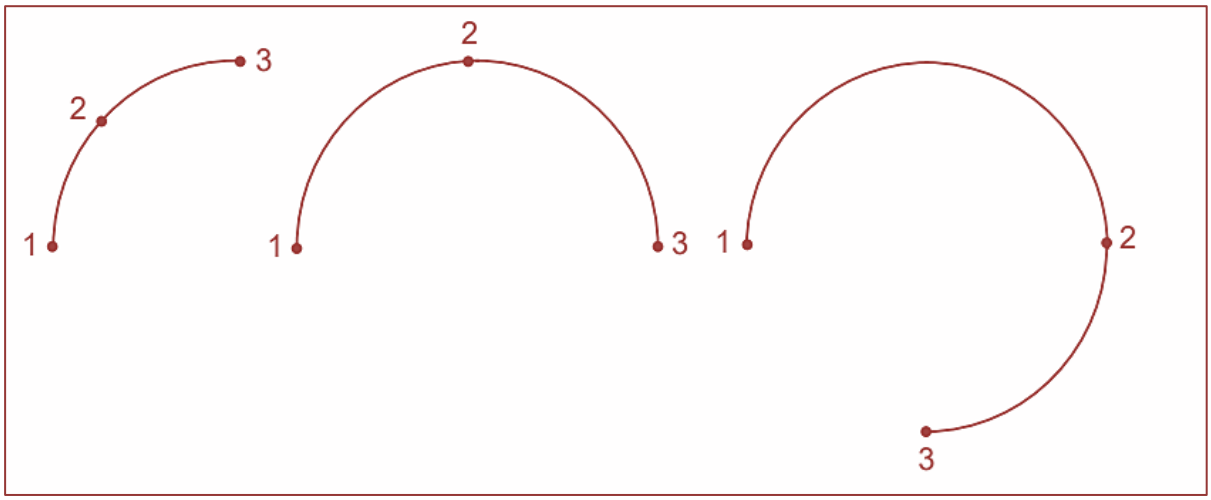
Figura 5.7 - Ilustração do movimento *Linear Motion Type* (MOVL)



Fonte: https://catalogo.yaskawa.com.br/, 2023.

O *Circular Motion Type* (MOVC) é um movimento no qual o robô pode mover o seu TCP em uma trajetória circular simétrica (seguindo uma trajetória parabólica ou descrevendo um polinômio de segundo grau). Para isso, basta gravar 3 pontos consecutivos com o movimento MOVC. Os 3 pontos permitem ao controlador determinar o plano, raio e centro do círculo interpolado. Cada ponto MOVC a mais que for gravado desnecessariamente fará com que o controlador recalcule todos os dados anteriormente mencionados. A Figura 5.8 ilustra este tipo de movimento.

Figura 5.8 - Ilustração do movimento *Circular Motion Type* (MOVC)



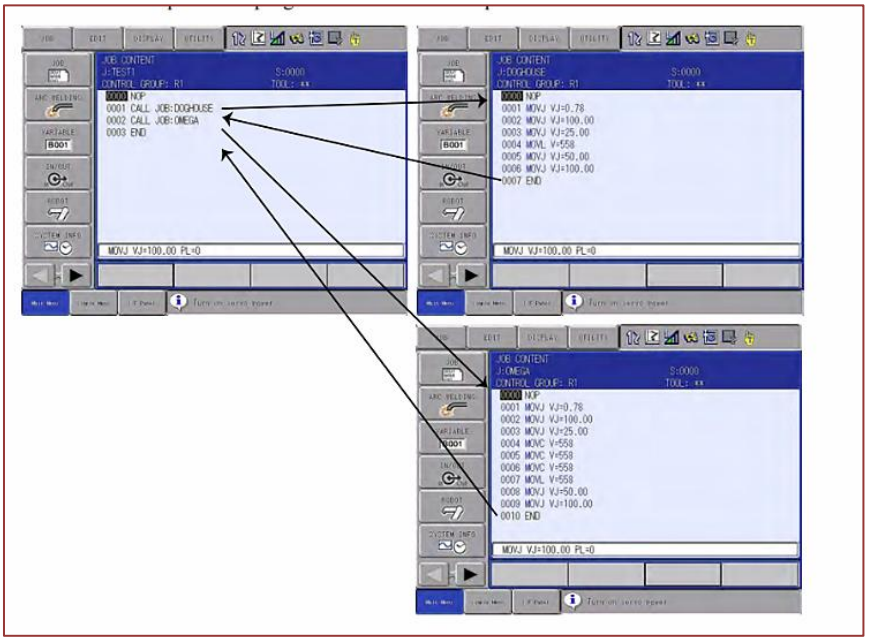
Fonte: YASKAWA, 2014.

5.2.6. INSTRUÇÕES DE CONTROLE

– CALL e RET

A instrução CALL permite que um programa chame outro como uma sub-rotina. O programa que chama o outro é chamado de “Pai”. O programa chamado é o “filho”, que pode ser finalizado com a instrução RET ou pelo fim do programa (linha END). O controlador continuará executando o programa pai com a próxima linha após a instrução CALL ser executada. Programas podem ser conectados criando um programa Pai com instruções CALL para cada programa Filho na ordem que eles devem ser executados. A Figura 5.9 demonstra o uso das instruções CALL e RET.

Figura 5.9 - Exemplo do uso das instruções CALL e RET



Fonte: Yaskawa, 2014.

– JUMP*LABEL e *LABEL

A instrução JUMP*LABEL funciona somente com a instrução *LABEL. Juntas, elas instruem o controlador a ir de uma linha com a instrução JUMP*LABEL para outra com a mesma função *LABEL. Essas instruções podem ser utilizadas para pular uma parte do programa ou criar um loop infinito. Um label automaticamente inclui um asterisco (*).

A instrução JUMP é utilizada para pular/omitir linhas ou para ignorar uma parte de um código. A instrução JUMP*LABEL precisa preceder a instrução *LABEL. No exemplo do trecho de código a seguir, o controlador pulará as linhas 6, 7 e 8, encontrando o label correspondente na linha 9, e continuará a execução na linha 10 (se a condição for verdadeira); caso contrário, o controlador executará todas as linhas.

```
0000 NOP
0001 MOVJ VJ=0.78
0002 MOVJ VJ=100.00
0003 MOVJ VJ=25.00 PL=0
0004 MOVL V=675
0005 JUMP *LABEL1 IF IN#(4) = ON
0006 MOVL V=675
0007 MOVL V=675
0008 MOVL V=675
0009 *LABEL1
0010 MOVL V=675
0011 MOVJ VJ=25.00
0012 MOVJ VJ=100.00
0013 END
```

A condição IF é uma condição que diz se a instrução de controle a ela atrelada será executada. Ela pode ser adicionada na Linha *de Buffer* (linha temporária) para uma nova instrução no lado dos endereços, ou para uma instrução existente no lado das instruções. A condição IF só pode ser acessada pela tela de detalhes das seguintes instruções de controle: • *JUMP* • *CALL* • *RET* • *PAUSE* • *ABORT* As condições disponíveis para a função IF são: • IN# - verifica o status de uma Input (ON ou OFF); • IG# - verifica o status de um input group (= ou <>; 0 a 255).

No exemplo do trecho de código a seguir, temos um exemplo para cada uso da instrução IF com várias instruções de controle:

```
0000 NOP
0001 CALL JOB:START
0002 CALL JOB:A IF IN#(1) = ON
0003 CALL JOB:B IF IN#(2) = ON
0004 JUMP*LABEL1 IF IG#(1) = 0
0005 SET B004 2
0006 *LABEL2
0007 MOVL V=558
0008 MOVL V=558
0009 MOVL V=558
0010 DEC B004
0011 JUMP *LABEL2 IF B004 > 0
0012 MOVL V=558
0013 *LABEL1 0014
MOVJ VJ=25.00
0015 MOVJ VJ=100.00
0016 PAUSE IF IN#(7) = OFF
0017 END
```

5.2.7. VARIÁVEIS E INSTRUÇÕES ARITMÉTICAS

Uma variável aritmética é um elemento de codificação capaz de armazenar um valor numérico, geralmente inteiro ou real. Estas variáveis são normalmente utilizadas para armazenar valores de controle do robô. O tipo da variável determina os valores que a variável pode receber. A Tabela 4.1 apresenta os tipos, endereços, faixa de utilização e aplicações destas variáveis.

Tabela 5.1 - Tipos, endereços, faixa de utilização e aplicações das variáveis aritméticas

Type	Addresses	Range	Application
Integer	I000-I099	-32768 to +32767	Counting
Double Precision	D000-D099	-2147483649 to +2147483647	Distance (Microns)
Real	R000-R099	±3.400000 E+38 (minimum/maximum) ±9.999999E-38 (values between -1 and +1)	Higher Arithmetic Calculations (DIV, SQRT, SIN, COS, ATAN, etc.)
Byte	B000-B099	0-255	I/O Communication

Fonte: Yaskawa, 2014.

Uma Instrução Aritmética é aquela utilizada com variáveis aritméticas, ou seja, para cálculos ou atribuições de valores de maneira geral. O menu ARITH do INFORM LIST contém uma vasta seleção de instruções aritméticas comumente utilizadas. A Tabela 4.2 apresenta as instruções usadas para contadores, começando por um valor inicial, e limpando o conteúdo de uma sequência de variáveis para o padrão, zero.

Tabela 5.2 - Instruções para contadores: funcionalidade, formato, exemplo e resultado obtido

	Function	Format	Example	Result
INC	Adds one to the contents of specified address.	INC B<address> INC I<address> INC D<address>	INC B023 INC I023 INC D023	If contents = 6 increments to 7.
DEC	Subtracts one from the contents of specified address.	DEC B<address> DEC I<address> DEC D<address>	DEC B023 DEC I023 DEC D023	If contents = 6 decrements to 5.
SET	Sets Data 1 equal to Data 2.	SET <Data 1> <Data 2>	SET I012 3	I012 becomes a constant of 3.
CLEAR	Clears range from address to specific value.	CLEAR <address> CLEAR <address>	CLEAR I007 4 CLEAR I007 ALL	Clears I007, 8, 9, 10 Clears from I007 on. (I007-I099)

Fonte: Yaskawa, 2014.

A instrução INC incrementa o valor armazenado em uma variável (+1). Por exemplo, se I012 tem 6 no seu valor, após a execução da instrução INC012, ele terá 7 em seu valor. A instrução INC só pode ser utilizada com as variáveis tipo B, I e D. O trecho de código a seguir exemplifica isto.

```
0013 MOVJ VJ=25.00
0014 MOVJ VJ=100.00
0015 INC I012
0016 END
```

A instrução DEC decrementa o valor armazenado em uma variável (-1). Por exemplo, se I022 tem 10 no seu valor, após a execução da instrução DEC I022, ele terá 9 em seu valor. A instrução DEC só pode ser utilizada com as variáveis do tipo B, I e D. A instrução DEC deve ser usada para alcançar uma meta de produção. Uma variável é selecionada para armazenar o valor necessário de peças a ser produzidas. Ela será decrementada cada vez que uma peça for produzida. Para melhores resultados, faça o decremento somente no final do programa de produção. O trecho de código a seguir exemplifica isto.

```
0004 MOVL V=725
0005 MOVL V=725
0006 MOVL V=725
0007 MOVL V=725
0008 DEC I022
0009 PAUSE IF I022 <= 0
0010 END
```

A instrução SET armazena qualquer valor em uma variável. Isso permite que o programa determine valores iniciais para variáveis sem depender da entrada manual de valores. Pode também ser usada para zerar uma variável individual; se for zerar mais de uma variável consecutiva, utilize a função CLEAR. O trecho de código a seguir exemplifica isto.

```
0004 MOVJ VJ=25.00
0005 MOVJ VJ=50.00
0006 SET B020 15
0007 SET I099 0
0008 END
```

A instrução CLEAR permite que variáveis consecutivas sejam zeradas. A primeira variável da sequência procede a instrução CLEAR. No trecho de código a seguir, CLEAR I007 4 zera o conteúdo de 4 variáveis, I007, I008, I009 e I010.

```
0004 MOVJ VJ=25.00
0005 MOVJ VJ=100.00
0006 CLEAR I007 4
0007 CLEAR B000 ALL
0008 END
```

5.2.8. INSTRUÇÕES DE ENTRADA E SAÍDA - I/O

Um sistema robótico raramente trabalha sem interação com outros dispositivos. O controlador deve se comunicar com equipamentos externos, como fontes de solda, posicionadores, dispositivos, sensores. A comunicação pode ser feita através do uso de Entradas e Saídas Universais (I/O). A linguagem INFORM III possui instruções de I/O digitais e analógicas, tais como a DOUT, WAIT e DIN.

A instrução DOUT com a tag OT#( ) opera apenas uma saída universal. É utilizado a qualquer hora que um dispositivo, como uma lâmpada, grampo, garra, deve ser ligado ou desligado. O trecho de código a seguir exemplifica o uso da função DOUT sendo utilizada com uma única saída universal

```
0003 DOUT OT#(3) ON
0004 MOVL V=2000
0005 MOVL V=2000
0006 DOUT OT#(3) OFF
```

A instrução WAIT é usada quando o programa do robô depende de um sinal de uma chave switch, sensor, PLC, etc. Isso abrange IN#, OT#, IG#, OG#, IGH#, OGH# (universais) e SIN#, SOUT# (específicos), assim como outras variáveis *Byte*. Um *TIMER* pode ser usado com *WAIT.* Quando o timer é utilizado, o robô espera que a condição seja verdadeira ou até que o tempo se esgote, o que vier primeiro. A seguir, todas as possibilidades da instrução WAIT:

- WAIT IN#(3)=ON [espera até que a IN#3 fique ON]
- WAIT IG#(1)=255 [espera até que o IG#(1) = 255]
- WAIT IN#(5)=OFF T=3.25 [espera que a IN#5 seja desligada por 3.25s continuando para a próxima linha].

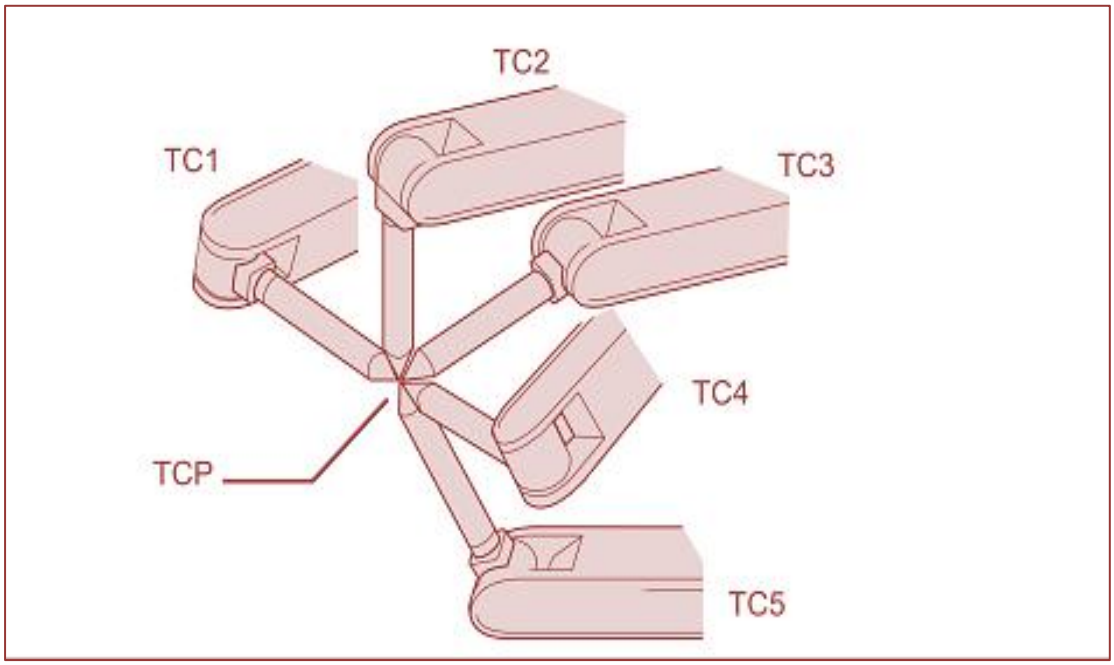
A instrução DIN (Digital Input) é usada para armazenar o estado de I/O em uma variável Byte. As informações de I/O são recebidas de um CLP, chave *thumbwheel* (chave hexadecimal) ou qualquer outro dispositivo de entrada. Isso inclui I/O universais: IN#, OT#, IG#, OG#, IGH# e OGH# ou específicos: SIN# e SOUT#, assim como qualquer outra variável Byte. Quando a instrução DIN é executada, o controlador verifica o estado de uma entrada ou saída e armazena o valor binário na variável Byte. Esse valor será entre 0 e 255, conteúdo este que deverá ser utilizado posteriormente no programa: 0001 DIN B005 IG#(1)

5.2.9. CALIBRAÇÃO DE FERRAMENTA

A calibração da ferramenta de robôs é fundamental para a melhoria de desempenho de posicionamento da ferramenta, ou atuador final, garantindo precisão necessária para a realização de determinadas tarefas. A calibração consiste na estimativa de parâmetros cinemáticos do mecanismo

através de diversas medições da posição da ferramenta que pode fornecer resultados de posição e/ou orientação. Existem dois métodos de calibrar a ferramenta para o robô Yaskawa o método manual e automático. A Figura 5.10 demonstra as posições para a maneira automática da calibração.

Figura 5.10 - Posições para calibração automática



Fonte: Yaskawa, 2014.

No método manual basta digitar o tamanho da ferramenta utilizada nas coordenadas X,Y,Z. Geralmente o fabricante da ferramenta informa seu dimensional em seu manual.

5.3. ROBÔ MANIPULADOR UNIVERSAL (UR)

O Robô Manipulador Universaol, ou Universal Robots (UR), é do tipo conhecido como Robô Colaborativo (Collaborative Robot - COBOT), que é um robô que permite a interação entre humanos e robôs em um espaço compartilhado ou onde estes estão próximos um ao outro, portanto possui todo um sistema de controle e segurança prevendo esta interação com seres humanos.

5.3.1. PARTES E LIGAÇÃO DO ROBÔ UR-COBOT

O robô UR-COBOT é formado pelo controlador, o manipulador e a interface de programação. A Figura 5.11 apresenta uma imagem do robô.

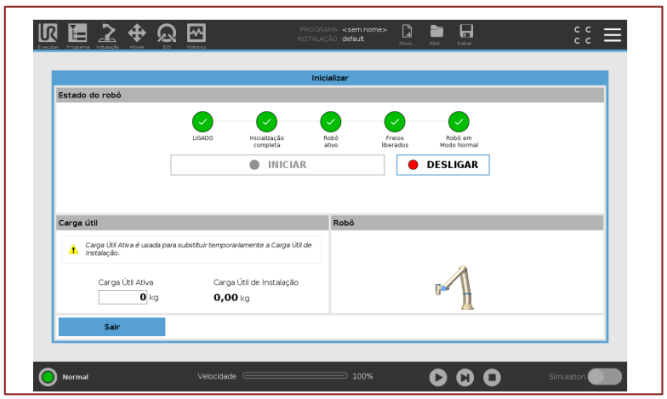
Figura 5.11 - Robô Manipulador Universal Colaborativo – UR COBOT

Fonte: Universal, 2023.

Na parte inferior esquerda da tela do controlador (Figura 5.12), o ícone Inicializar indica o status do braço do robô utilizando cores:

- Vermelho: O Braço do robô está em um estado parado.
- Amarelo: O braço do robô está ligado, mas não está preparado para funcionar normalmente.
- Verde: O braço do robô está ligado e preparado para funcionar normalmente.

Figura 5.12 - Tela do controlador de inicialização do Robô



Fonte: UNIVERSAL, 2023.

5.3.2. MODOS DE UTILIZAÇÃO E PROGRAMAÇÃO

– Modos Operacionais

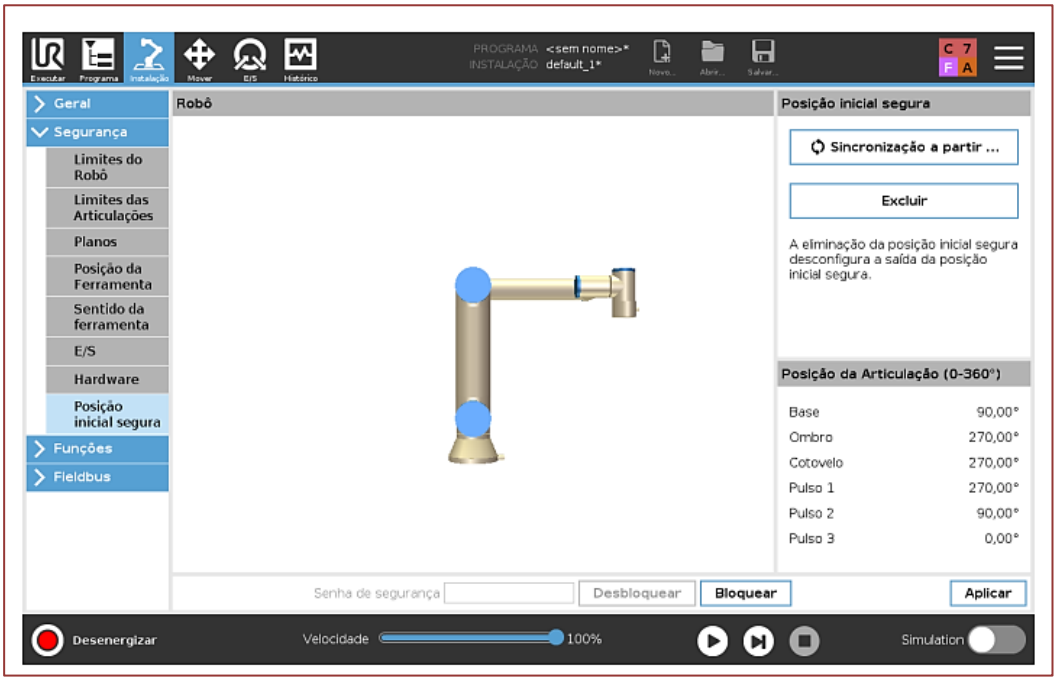
Os Modos Operacionais são ativadas quando você configurar um Dispositivo de Ativação de Posição 3 ou definir uma senha. Modo Automático uma vez ativado, o robô só

pode executar as tarefas pré-definidas. A guia Mover e o modo de Condução Livre estão indisponíveis. Você não pode modificar ou salvar programas e instalações. No Modo Manual, depois de ativado, você programar que o robô está usando a guia Mover, o Modo de Condução Livre e o Botão Deslizante de Velocidade. Você pode modificar ou salvar programa e instalações.

– Posição Inicial Segura

A Posição inicial segura (Figura 5.13) é uma posição de retorno definida usando a posição inicial definida pelo usuário. As E/S iniciais seguras estão ativas quando o braço do robô está na posição inicial segura e uma E/S inicial segura é definida. O braço do robô está na posição inicial segura se as posições da articulação estão nos ângulos da articulação especificados ou em um múltiplo de 360 graus.

Figura 5.13 - Definindo a posição inicial segura do Robô

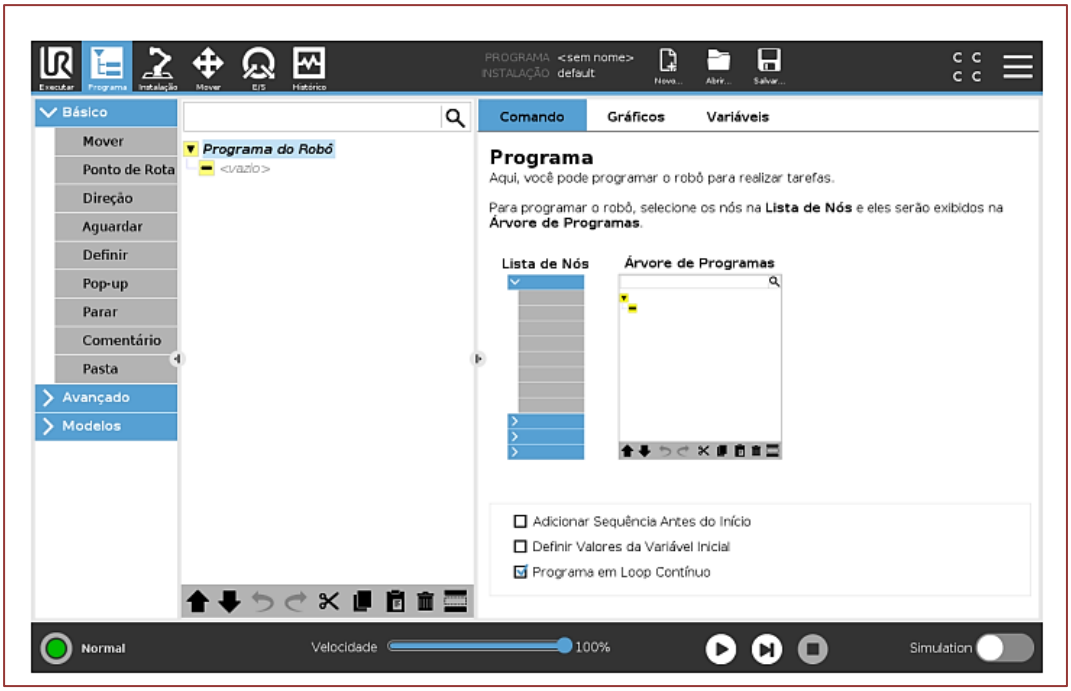


Fonte: UNIVERSAL, 2023.

– Árvore de Programas

Ao tocar em Comando, você adiciona “Nós” do programa à Árvore do programa (Figura 5.14), e configura a funcionalidade dos nós do programa adicionados no lado direito da tela. Uma Árvore do programa vazia não pode ser executada. Os programas contendo os nós do programa configurados incorretamente também não podem ser executados. Os nós do programa inválidos são destacados em amarelo para indicar o que deve ser consertado antes que programa possa ser executado.

Figura 5.14 - Árvores de programas



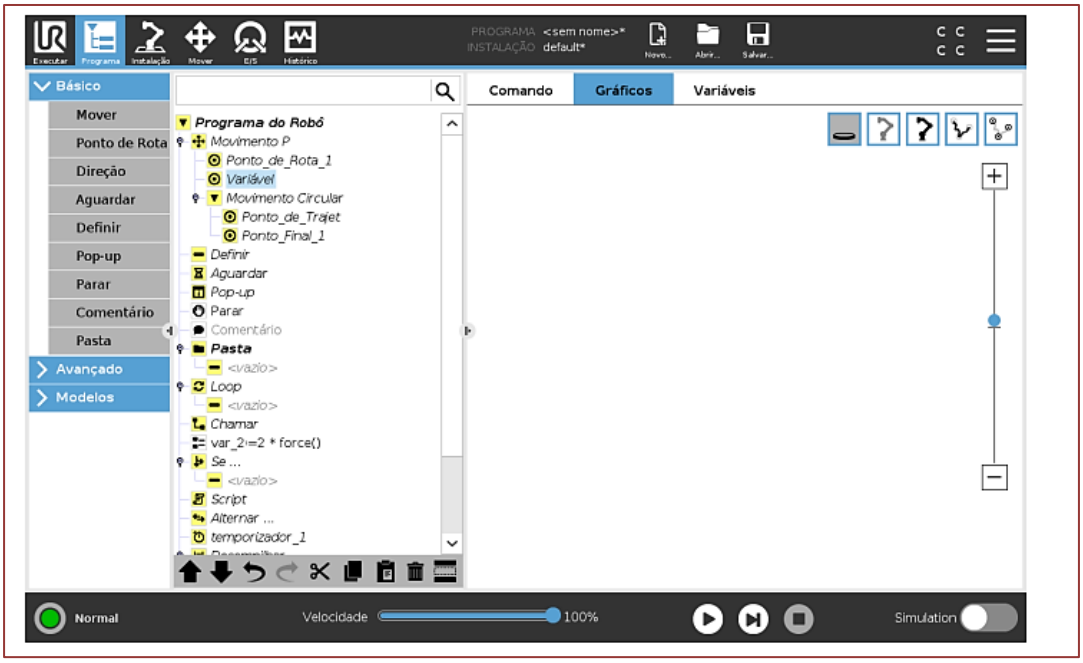
Fonte: Universal, 2023.

Quando o programa é executado, o nó do programa que está sendo executado atualmente é indicado por um ícone pequeno ao lado do nó. Além disso, o caminho de execução é destacado com uma cor azul. Pressionar o ícone no canto do programa fará com que ele rastreie o comando que está sendo executado.

– Representação gráfica do programa

A representação gráfica do programa apresentar a posição e orientação atual do robô. O caminho do TCP é mostrado na visualização 3D, com segmentos de movimentação em preto e segmentos de união (transições entre segmentos de movimentação) mostrados em verde. Os pontos verdes especificam as posições do TCP em cada um dos pontos de rota no programa. O desenho 3D do braço do robô mostra a posição atual do braço, e sua sombra mostra como o braço do robô pretende chegar ao ponto de rota selecionado do lado esquerdo da tela. Se a posição atual do TCP do robô se aproximar de um plano de segurança ou de acionamento, ou a orientação da ferramenta do robô estiver perto do limite de orientação da ferramenta. (Figura 5.15).

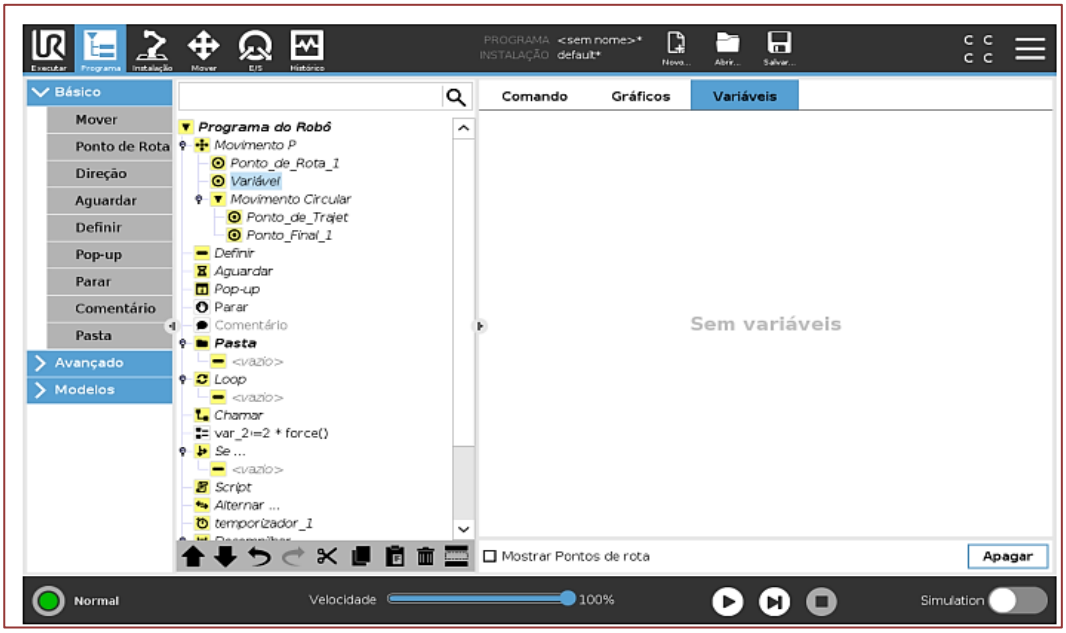
Figura 5.15 - Representação gráfica: Posição do TCP



Fonte: Universal, 2023.

A guia Variáveis mostra os valores ativos das variáveis no programa em execução e mantém uma lista de variáveis entre execuções do programa. Ela aparece somente quando tem informações para exibir. Variáveis de pontos de rota serão mostradas na lista, se Mostrar Pontos de rota estiver ativado. e-Series II-40 Versão 5 (Figura 5.16).

Figura 5.16 - Tela das variáveis ativas

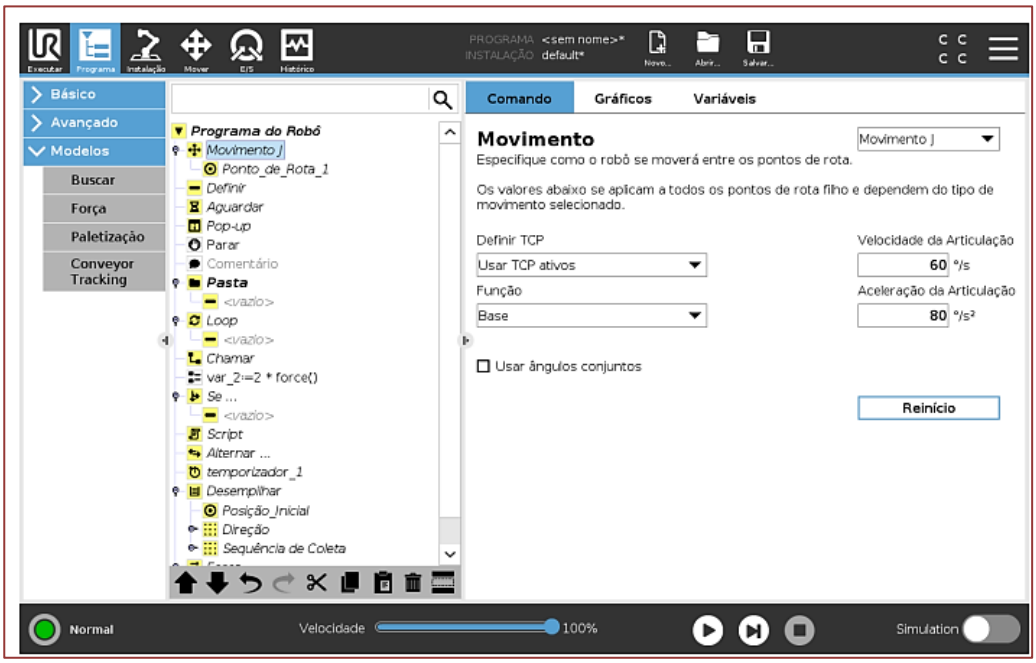


Fonte: Universal, 2023.

– Comando Mover

O comando Mover controla a movimentação do robô por meio de pontos de rota subjacentes. Os Pontos de Rota precisam estar sob o comando Mover. O comando Mover define a aceleração e a velocidade em que o braço do robô se moverá entre esses pontos de rota (Figura 5.17).

Figura 5.17 - Comando MOVER de movimentação do robô



Fonte: Universal, 2023.

Em relação aos tipos de movimentos, é possível selecionar um de três tipos de movimentos: Movimento J, Movimento L e Movimento P. Cada tipo será explicado a seguir.

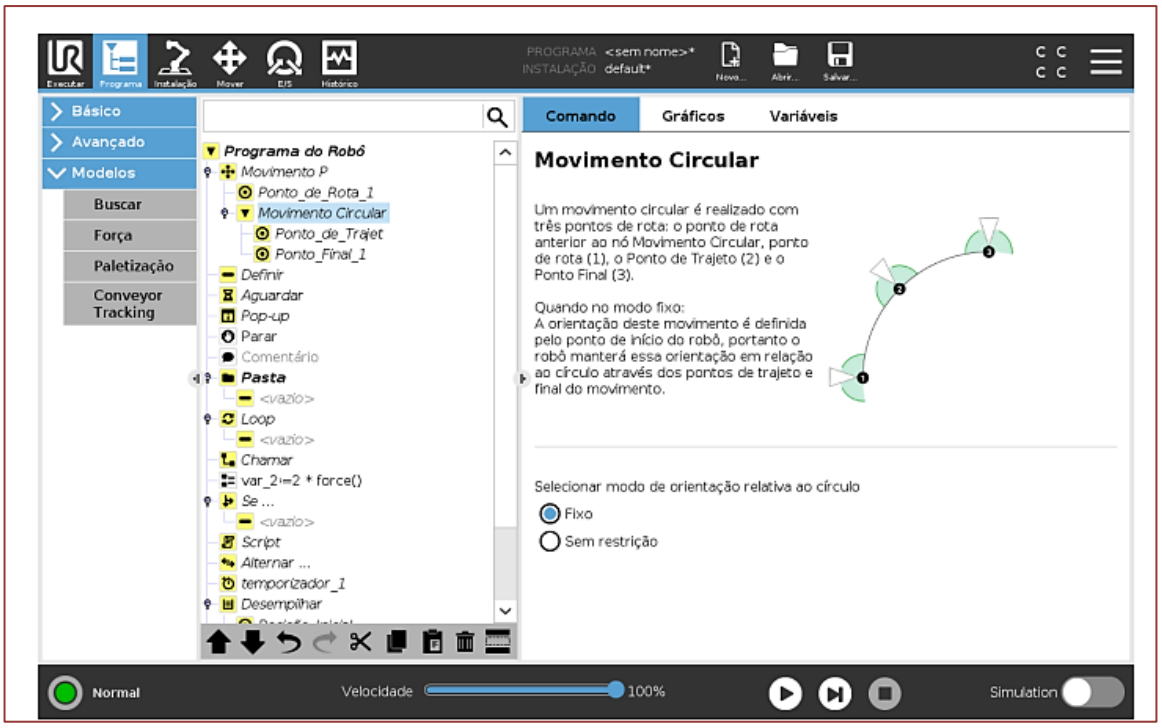
O Movimento J faz movimentos que são calculados no espaço de articulação do braço do robô. As articulações são controladas para terminar os seus movimentos ao mesmo tempo. Esse tipo de movimento resulta em um caminho curvo para a ferramenta. Os parâmetros compartilhados que se aplicam a esse tipo de movimento são a velocidade máxima da articulação e a aceleração da articulação, especificados em graus/s e graus/s 2, respectivamente. Se for desejado que o braço do robô se mova rapidamente entre pontos de rota, desconsiderando o caminho da ferramenta entre esses pontos de rota, esse tipo de movimento será a opção preferencial.

O Movimento L move o Ponto Central da Ferramenta (TCP) linearmente entre pontos de rota. Isso significa que cada articulação executa uma movimentação mais complicada para manter a ferramenta em um caminho de linha reta. Os parâmetros compartilhados que podem ser definidos para esse tipo de movimento são a velocidade desejada da ferramenta e a aceleração da ferramenta especificadas em mm/s e mm/s^2, respectivamente, além de um recurso.

O Movimento P move a ferramenta linearmente com velocidade constante com uniões circulares e é destinado a algumas operações do processo, como colar e descartar. O tamanho do raio de união é por padrão um valor compartilhado entre todos os pontos de rota. Um valor menor fará o caminho se tornar mais angular, enquanto um valor mais alto fará o caminho mais suave. Enquanto o braço do robô estiver se movendo pelos pontos de rota com velocidade constante, a caixa de controle do robô não poderá esperar por uma operação de E/S ou uma ação do operador. Fazer isso poderá parar a movimentação do braço do robô ou causar uma parada de proteção.

O Movimento circular (Figura 5.18) pode ser adicionado a um Movimento P para fazer um movimento circular. O robô inicia o movimento a partir de sua posição atual ou ponto de início, movimenta-se através de um Ponto de Trajeto especificado no arco circular, e um Ponto Final, que completa o movimento circular. Um modo é usado para calcular a orientação da ferramenta, através do arco circular. O modo pode ser: – Fixo: apenas o ponto de início é usado para definir a orientação da ferramenta – Sem restrição: o ponto de início se transforma em Ponto Final para definir a orientação da ferramenta Parâmetros compartilhados.

Figura 5.18 - Movimento circular do robô



Fonte: Universal, 2023.

5.4 ROBÔ EPSON SCARA

O robô do tipo SCARA foi desenvolvido pela primeira vez nos anos 80 no Japão, e o nome SCARA significa *Selective Compliance Assembly Robot Arm*. Apesar do nome extenso, na pratica ele possui um braço articulado de 2 eixos que, em certa medida, imita o braço humano, embora ele opere em um único plano numa movimentação do tipo XYZ.

5.4.1. PARTES DO ROBÔ EPSON SCARA

O Robô Epson SCARA é formado por controlador e manipulador e a programação é feita através do computador com software específico do fabricante. Sua construção permite a ele se estender ou se retrair em áreas confinadas.

Quando se trata de operações de *pick-and-place* ou montagem, muito comuns no Polo Industrial de Manaus, os robôs SCARA são a opção perfeita, já que estas aplicações demandam alta velocidade e alta precisão. Neste cenário, eles são capazes de operar com altas acelerações e alta velocidade, tanto em ambientes agressivos quanto limpos.

A Figura 5.19 apresenta o Robô Epson SCARA, o manipulador e o controlador.

Figura 5.19 - Manipulador e controlador do Robô Epson SCARA

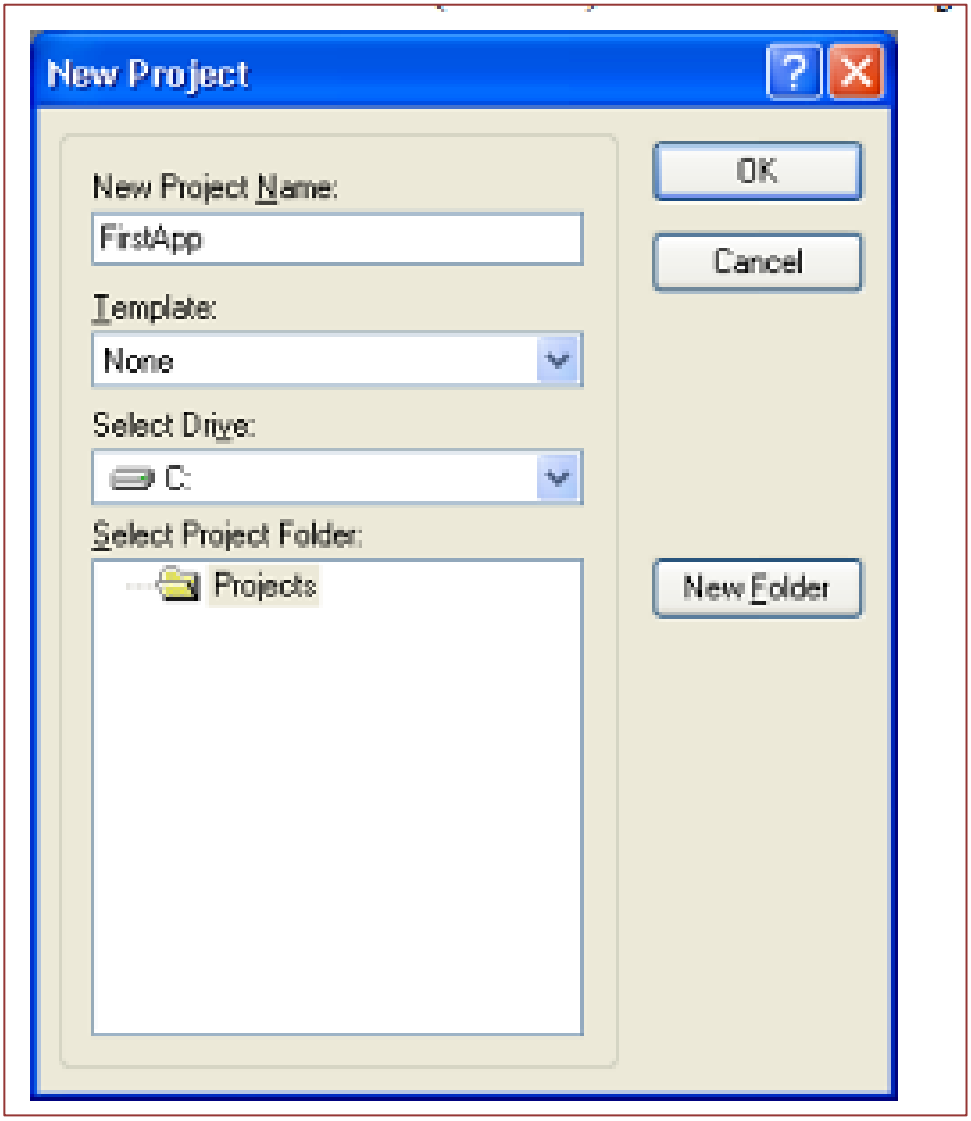
Fonte: EPSON, 2023.

5.4.2. PROGRAMAÇÃO DO ROBÔ EPSON SCARA

– Criando um programa

Depois de instalar o controlador, o robô e o software EPSON RC+ 7.0 no Controlador de Robô RC700, siga estas instruções para criar um programa aplicativo simples para se familiarizar com o ambiente de desenvolvimento do EPSON RC+ 7.0. 1. Inicie o EPSON RC+ 7.0 Dê um clique duplo no ícone do robô do EPSON RC+ 7.0 na área de trabalho. (Figura 5.20).

Figura 5.20 - Criando um novo programa (projeto)

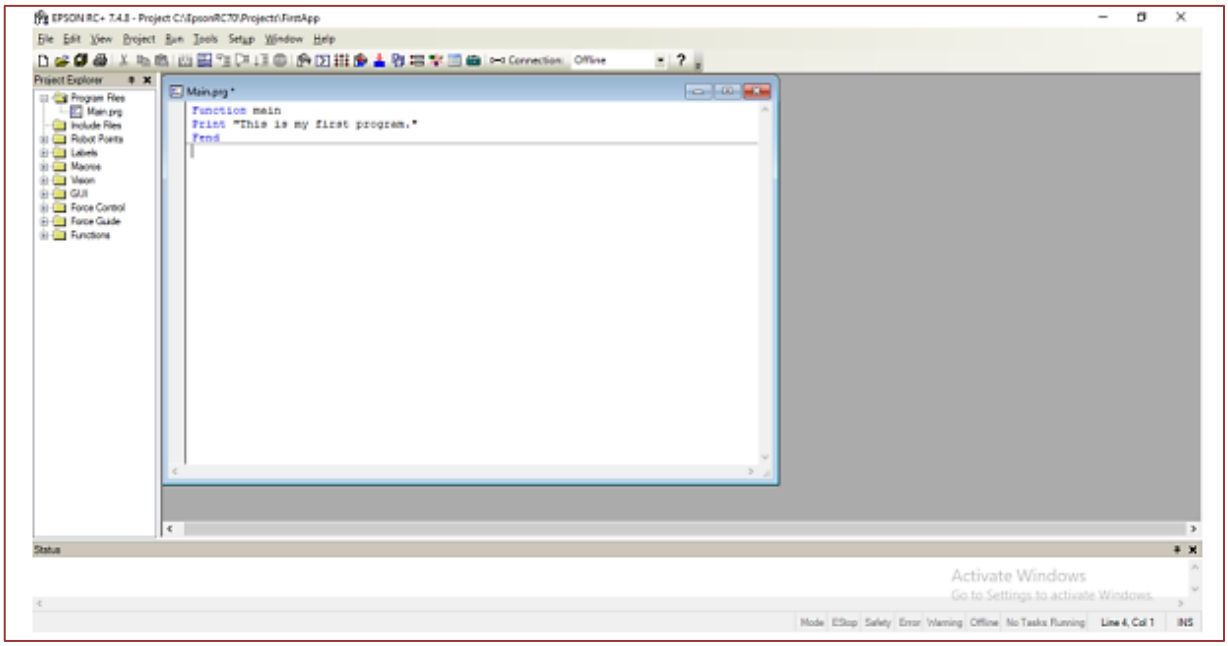


Fonte: EPSON, 2023.

Digite um nome para o projeto na caixa [New Project Name] (Nome do novo projeto). P.ex. FirstApp (3) Clique em OK para criar o novo projeto. Quando um novo projeto é criado, um programa chamado “Main.prg” é criado. Você verá a janela “Main.prg” aberta com um cursor piscando no canto superior esquerdo. Agora você está pronto para começar a entrar em seu primeiro programa.

Digite as seguintes linhas de programa na janela de edição “Main.prg”. Function main Print "*This is my first program*." Fend 4. Execute o programa, para isto, pressione F5 para executar o mesmo. A Figura 5.21 apresenta o resultado deste pequeno exemplo.

Figura 5.21 - Exemplo de primeiro programa



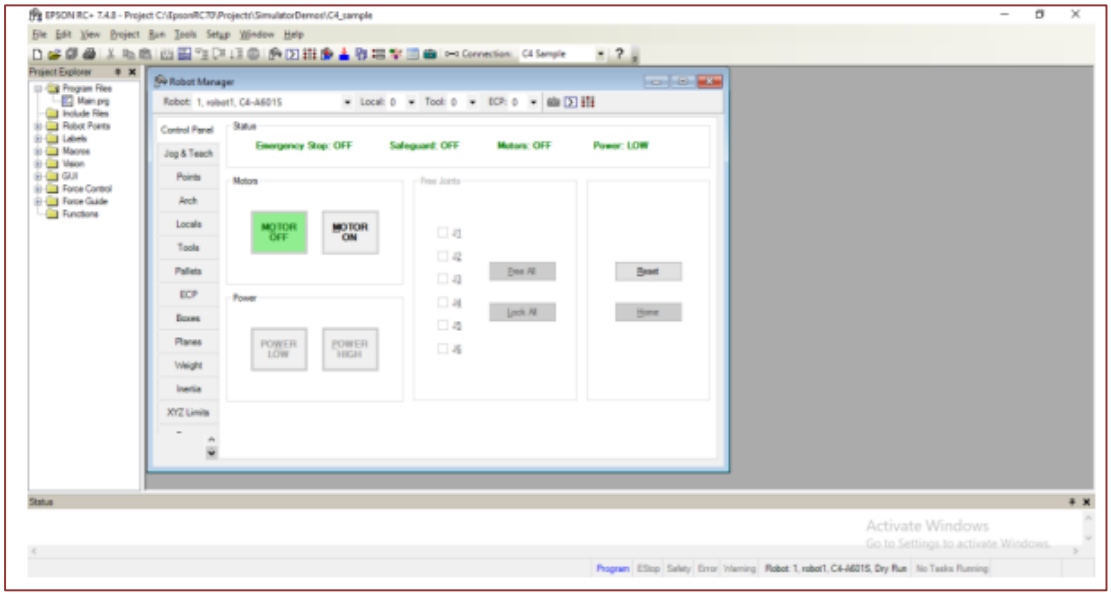
Fonte: EPSON, 2023.

Durante a construção do projeto, seu programa é compilado e vinculado. A comunicação é estabelecida com o controlador e os arquivos do projeto são enviados para ele. Se não houver erros durante a construção, a janela de execução aparecerá. A partir daí, clique no botão (Iniciar) na janela [Run] para executar o programa. Você deverá ver um texto similar ao seguinte exibido na janela [Status]: Task main started - All tasks stopped. Na janela [Run], você verá a saída da instrução de impressão.

– Ensinando os pontos ao robô

1) Certifique-se de que é seguro operar o robô, com o mesmo com o motor no estado off. Clique no botão (Gerenciador do robô) na barra de tarefas. Você verá a janela [Robot Manager] com a página [Control Panel] (Painel de controle) exibida (*Cf.* Figura 5.22).

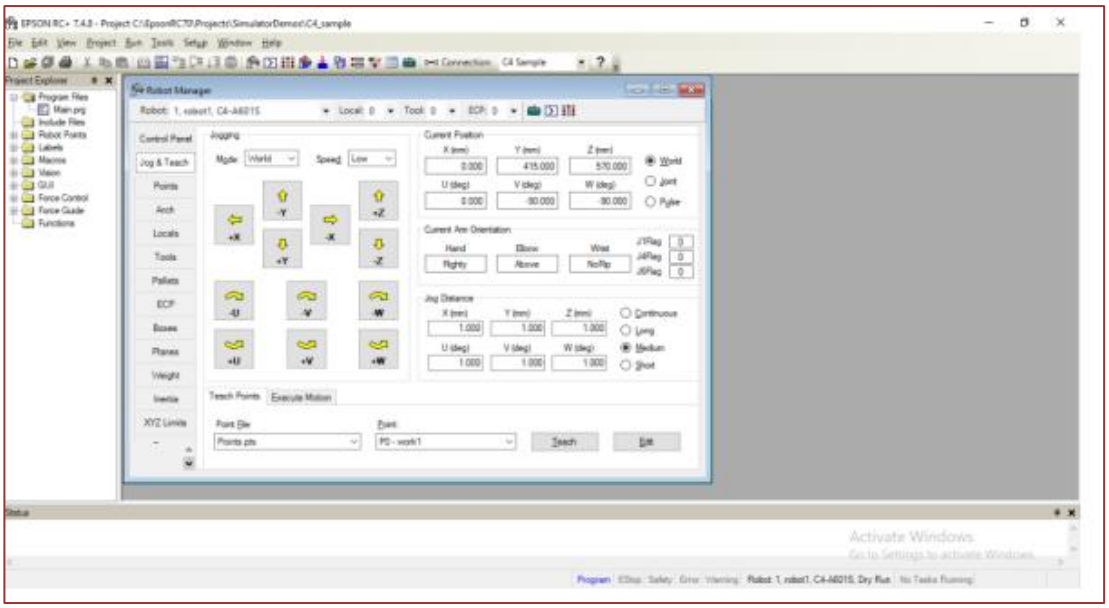
Figura 5.22 - Robô com motor off seguro para programar



Fonte: EPSON, 2023.

2) Clique no botão para ligar os motores do robô. Você será solicitado a confirmar a operação.
3) Responda (Sim) para continuar.
4) Clique na aba [Jog & Teach] (Deslocar e ensinar) (Cf. Figura 5.23)

Figura 5.23 - Tela Jog & Tech para definir os pontos de deslocamento do Robô



Fonte: EPSON, 2023.

5) Clique no botão (Ensinar) no canto inferior direito para ensinar o ponto P0. Você será solicitado a inserir um rótulo e a descrição do ponto. (6) Desloque o robô clicando no botão de deslocamento. Mantenha o botão pressionado para continuar deslocando. Solte quando o robô estiver aproximadamente na metade do caminho em direção à área de trabalho. (7) Desloque o robô para baixo clicando no botão <-Z>. (8) Agora mude o ponto atual para P1 selecionando P1 na lista suspensa Point (Ponto) ao lado do botão . (9) Clique no botão . Você verá uma mensagem de confirmação para ensinar o ponto. (10) Responda. (11) Clique no botão para deslocar o robô na direção +X. (12)Mude o ponto atual para P2 selecionando P2 na lista suspensa Point.

Como exemplo, insira três novas instruções Go no programa Main.prg como é mostrado no trecho de código a seguir:

```
Function main Print "This is my first program."
Motor On
Go P1
Go P2
Go P0
Fend
```

Depois, insira os comandos Power (Potência), Speed (Velocidade) e Accel (Aceleração) como é mostrado no trecho de código a seguir:

```
Function main Print "This is my first program."
Motor On
Power High
Speed 50
Accel 50, 50
Go P1
Go P2
Go P0
Fend
```

Em seguida, clique no botão na janela Run. O robô deverá ir para cada um dos pontos que você ensinou à metade da velocidade, aceleração e desaceleração. A instrução Power High (Potência alta) permite que seu programa execute o robô em alta potência (normal), o que, por sua vez permite que a velocidade e a aceleração sejam aumentadas.

5.5. ROBÔ MANIPULADOR STÄUBLI

O robô manipulador Stäubli é similar ao robô manipulador Yaskawa, sendo mais robusto e com maior carga de atuação, devido sua composição estrutural. Por conta disto, é um robô industrial muito utilizado para o processamento de peças metálicas, desde a usinagem até a carga e descarga de máquinas-ferramenta.

5.5.1. PARTES DO ROBÔ STÄUBLI

Como os demais, é formado por controlador e manipulador e a programação é feita através do computador com software específico do fabricante. O robô Stäubli foi projetado com braços robóticos compactos e fechados com várias opções de fixação (Figura 5.24).

Figura 5.24 - Manipuladores e Controlador do Robô Stäubli

Fonte: https://www.staubli.com, 2023.

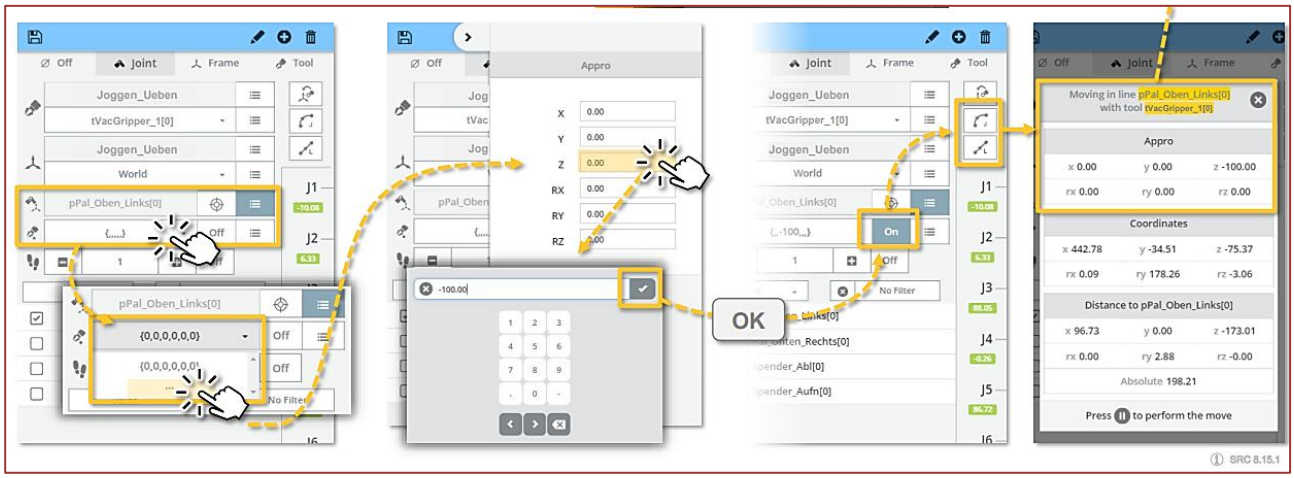
5.5.2. PROGRAMAÇÃO DO ROBÔ STÄUBLI

Devido a um atraso na entrega do equipamento, importado, este robô foi o menos utilizado na capacitação em robótica industrial avançada. Portanto, daremos um breve resumo de sua programação. Toda a programação é feita na plataforma *Stäubli Robotics Suite*, que pode se conectar à controladora para obter automaticamente os parâmetros do modelo de robô utilizado, evitando que tais parâmetros sejam colocados manualmente. A programação deverá sempre ser no modo *offline*, e assim como outros robôs similares, pode ser utilizado o TP (*Teach Pendant*).

- Função Aproach(aproximação)

Quando da definição dos pontos e velocidades para a trajetória do robô, pode ser gravado um ponto de aproximação gravando um valor próximo do ponto, utilizando a função *aproach*. Nesta etapa é utilizado o TP para gravar o ponto. A Figura 5.25 apresenta um exemplo do uso desta função.

Figura 5.25 - Uso da função Aproach

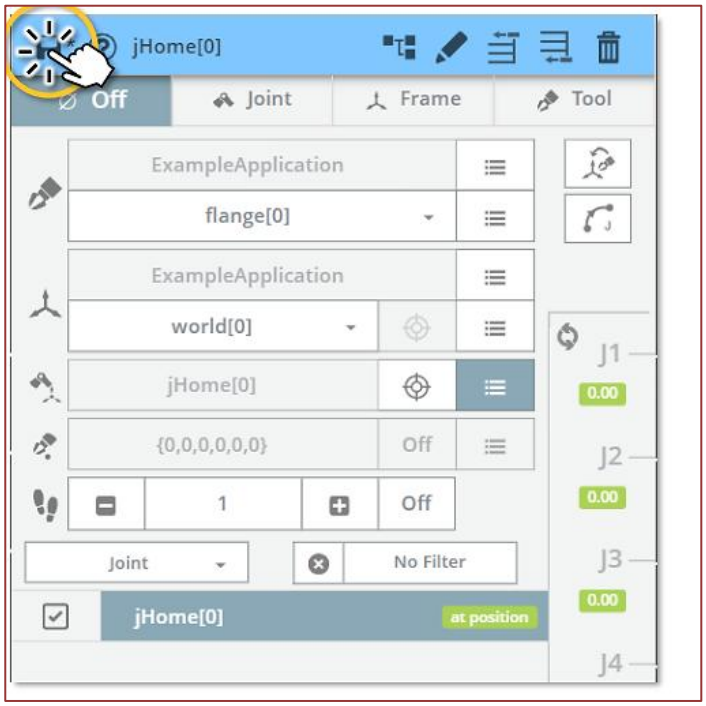


Fonte: Staubli, 2023.

– Salvando alterações.

A cada ponto de posição, tempo ou velocidade que for modificado, as alterações devem ser salvos para que sejam retidos após o aplicativo ser fechado. O modo Jog é indicado com um asterisco próximo ao ícone “Salvar”, e isto indica que foram feitas mudanças que ainda não foram salvas. Para salvar as alterações, toque no ícone Salvar – o asterisco desaparece. (vide Figura 5.26).

Figura 5.26 - Salvando as alterações

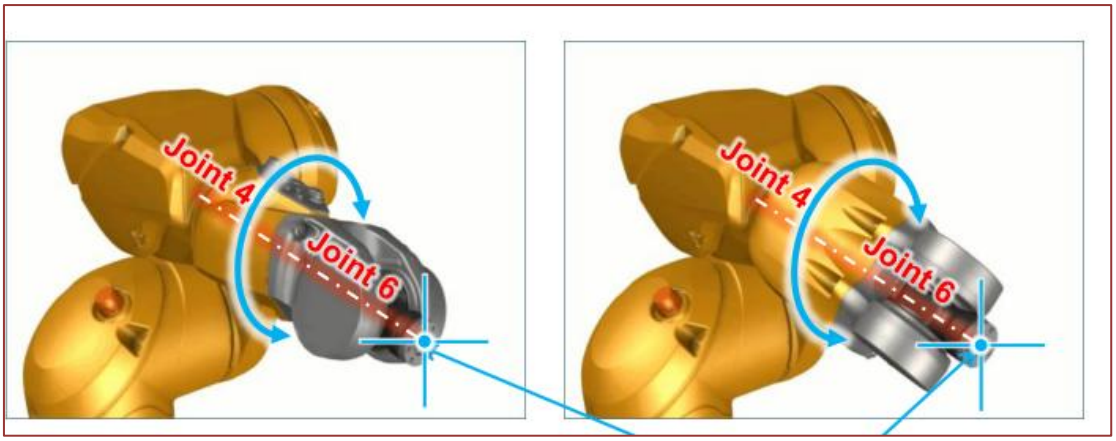


Fonte: Staubli, 2023.

– Singularidades

Singularidades são pontos no espaço que podem ser alcançados por uma distância infinita, e com isto geram um número de posição conjunta. A Figura 5.27 exemplifica um ponto singular.

Figura 5.27 - Pontos singulares



Fonte: Staubli, 2023.

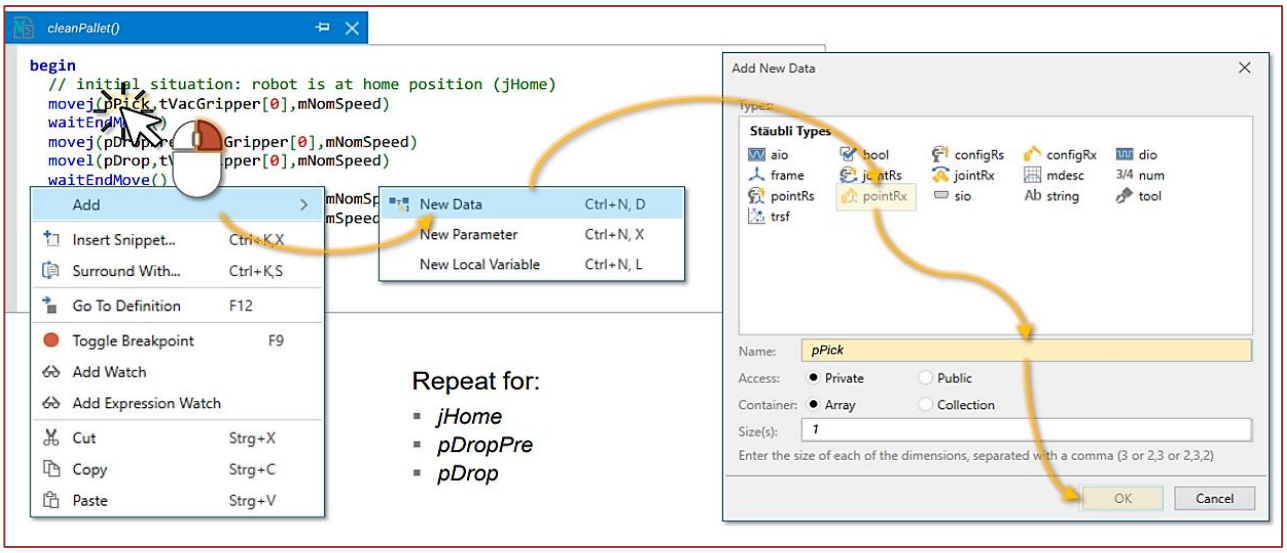
– Exemplo de programa

A seguir apresentamos um trecho de Código como exemplo de programação do robô stäubli. Basicamente, são um conjunto de instruções de movimentação do robô. O comando *movej* movimenta o ponto da ferramenta, tendo como parâmetros a variável ponto, tipo de movimento e velocidade, e o comando *waitEndMove* impõe uma espera do final do movimento anterior para que o próximo comando seja efetuado.

```
begin // initial situation: robot is at home position (jHome)
movej(pPick,tVacGripper[0],mNomSpeed)
waitEndMove()
movej(pDropPre,tVacGripper[0],mNomSpeed)
movel(pDrop,tVacGripper[0],mNomSpeed)
waitEndMove()
movel(pDropPre,tVacGripper[0],mNomSpeed)
movej(jHome,tVacGripper[0],mNomSpeed
```

A variável ponto pode ser adicionada ou definida seu tipo simplesmente colocando o curso sobre ela e seguindo o fluxo das telas conforme a Figura 5.28.

Figura 5.28 - Adicionar ou redefinir variável ponto



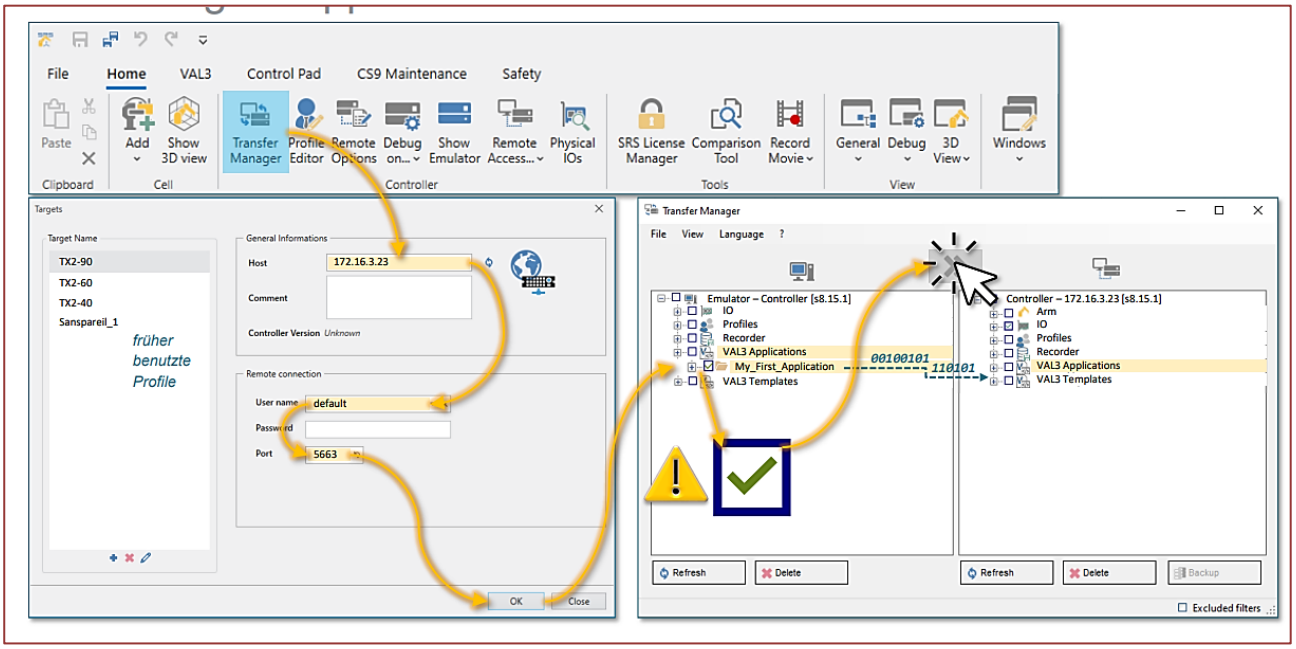
Fonte: Staubli, 2023.

– Transferindo a programação para o controlador

Uma vez concluído o programa na plataforma, de modo offline, ele deve ser transferido para o controlador. Esta etapa necessita configurar a rede (endereço IP) para a controladora e o equipamento utilizado para a programação (desktop ou notebook).

O programa pode ser transferindo seguindo o fluxo das telas conforme a Figura 5.29.

Figura 5.29 - Transferindo a programação para o controlador



Fonte: Staubli, 2023.

REFERÊNCIAS

[1] EPSON Robot. Manual Epson LS10-B Robot (Revision 6) Portuguese. Disponível em https://files.support.epson.com/far/docs/epson_ls10b_robot_manual_(r6)_portuguese_cpd60411.pdf, acesso em dez 2023.

[2] EPSON RC+ 7.0 Guia do usuário Gestão e desenvolvimento de projetos Ver.7.5 REV1. 606 páginas.

[3] UNIVERSAL Robots Academy,2023.Disponível em : https://www.universal-robots.com/br/academy/ Acesso em 10/08/2023.

[4] STAUBLI CS9 Programming Course - (16.03.2023) - © Staubli. Documentação privada e não publicada, com acesso somente mediante participação no treinamento.

[5] YASKAWA MOTOMAN. **Manual de Programação Básica DX200**. São Paulo-SP: Yaskawa do Brasil, 1ª Edição ,17 de junho de 2014, 183 páginas.

CAPÍTULO

06 Programação em CLP

Hugo Alves Velozo[1], Vanderson de Lima Reis[2]

6.1. CARACTERIZAÇÃO DA DISCIPLINA

Esta disciplina oferece uma abordagem abrangente à Programação em Controladores Lógicos Programáveis (CLP), destacando-se a utilização de linguagem ladder, blocos de função e texto estruturado. Os estudantes serão guiados na aquisição de habilidades técnicas fundamentais de programação de CLPs, explorando desde a representação gráfica intuitiva da lógica de controle até a abordagem modular para sistemas complexos. Uma ênfase especial será dada à integração de sistemas robotizados com CLPs e sistemas de visão, proporcionando aos alunos uma visão holística das tecnologias modernas de automação.

A metodologia adotada para ministrar desta disciplina será a utilizada no projeto, com maior enfoque para a Aprendizagem Baseada em Projetos (PBL), onde os estudantes serão desafiados a aplicar seus conhecimentos teóricos na resolução de problemas práticos seguindo a metodologia desenvolvida por Queiroz-Neto et al. (2015). Isso proporcionará uma experiência de aprendizado mais envolvente e aplicável, preparando os alunos para enfrentar os desafios reais encontrados na implementação de sistemas de controle e automação industrial. Ao final do curso, os alunos não apenas terão domínio das linguagens de programação em CLP, mas também estarão aptos a integrar eficientemente sistemas robotizados com CLPs e tecnologias de visão em projetos de automação avançada.

6.1.1. EMENTA

Nesta disciplina, será adotada a metodologia de ensino centrada na Aprendizagem Baseada em Projetos (PBL), proporcionando uma abordagem prática e aplicada aos estudantes. O curso iniciará com a conceituação de Controladores Lógicos Programáveis (CLPs), incluindo uma análise dos diferentes tipos disponíveis no mercado e suas respectivas aplicações industriais. Serão explorados casos reais e exemplos práticos para ilustrar a diversidade de implementações possíveis. Em seguida, serão discutidos os componentes modulares de um CLP, detalhando a arquitetura e as funcionalidades essenciais. Aprofundar-se-á no conhecimento da programação por meio do software GXWorks 2, com foco nas linguagens Ladder, Blocos de Função e Texto Estruturado. Além disso, será abordada a integração deste software com Factory IO para simulação realista, bem como a interação com sistemas robóticos e sistemas de visão, proporcionando aos alunos uma visão abrangente e prática da automação industrial moderna.

[1] Mestre em Automação e Controle. Professor da disciplina de Programação em CLP. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). E-mail: *hugo.velozo@ifam.edu.br.*
[2] Doutor em Engenharia Elétrica. Professor da disciplina de Programação em IHM. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). E-mail: *vanderson.reis@ifam.edu.br.*

6.1.2. OBJETIVOS

O objetivo da disciplina é capacitar os alunos na programação de Controladores Lógicos Programáveis (CLPs) para automação industrial de sistemas robotizados. Utilizando a metodologia de Aprendizagem Baseada em Projetos (PBL), o curso abrange conceitos de CLPs, tipos disponíveis, aplicações práticas e programação, com enfoque em linguagens como Ladder, Blocos de Função e Texto Estruturado. A formação inclui simulações realistas e exploração da interação com sistemas robóticos e de sistemas de visão, proporcionando uma preparação abrangente para enfrentar desafios na automação e robotização de ambientes industriais.

6.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 30 h

Carga Horária Prática: 30 h

Total: 60 h

6.2. CONTEÚDOS

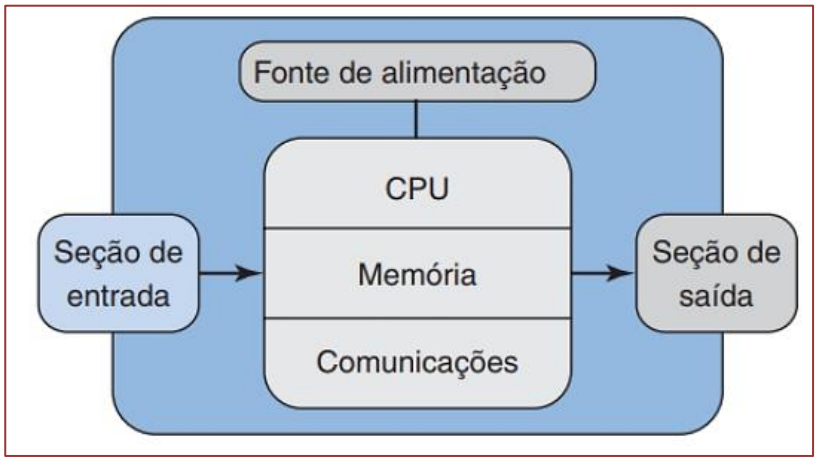
A seguir serão apresentadas breves introduções aos conteúdos ministrados durante a disciplina.

6.2.1. FUNDAMENTOS DE CONTROLADORES LÓGICOS PROGRAMÁVEIS - HARDWARE

Um Controlador Lógico Programável (CLP) é um computador usado para realizar funções de controle e automação em Ambiente Industrial tais como, operações condicionadas (operações lógicas), selos, intertravamentos, operações temporizadas, contagem, operações matemáticas, operações de controle, dentre outras (DA SILVA, 2016).

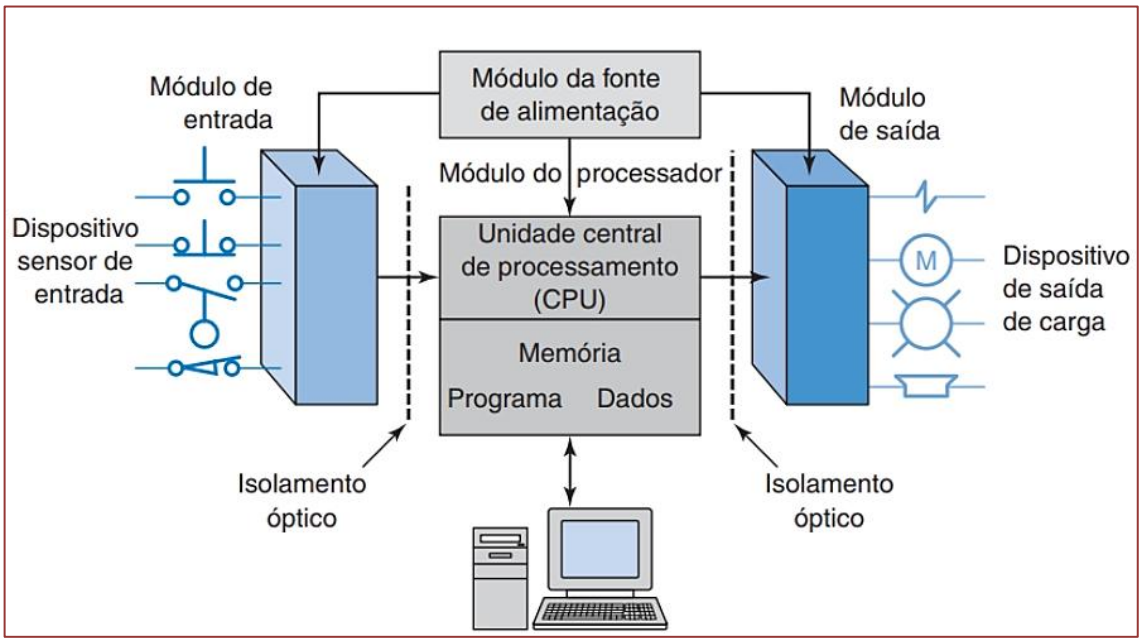
Os CLPs podem ser subdivididos em duas categorias principais quanto a forma da distribuição das entradas e saídas (no inglês, *I/O inputs* e *outputs*): os fixos e os modulares (PRUDENTE, 2015). Nos CLPs de I/O fixas, as entradas e saídas são incorporadas a estrutura do CLP sem possuir unidades removíveis, conforme ilustrado na Figura 6.1. Por outro lado, nos CLPs modulares, o CLP é dividido em módulos, ou seja, partes removíveis que formam a estrutura completa do CLP: base, fonte, unidade central de processamento (CPU, *central processing unit*), módulo de comunicação e módulos de entradas e saídas (WHITE, 2023). A Figura 6.2 ilustra genericamente a ideia de um CLP modular.

Figura 6.1 - Diagrama de um CLP com entradas/saídas fixas



Fonte: Petruzella, 2014.

Figura 6.2 - Diagrama de CLP com entradas/saídas modulares

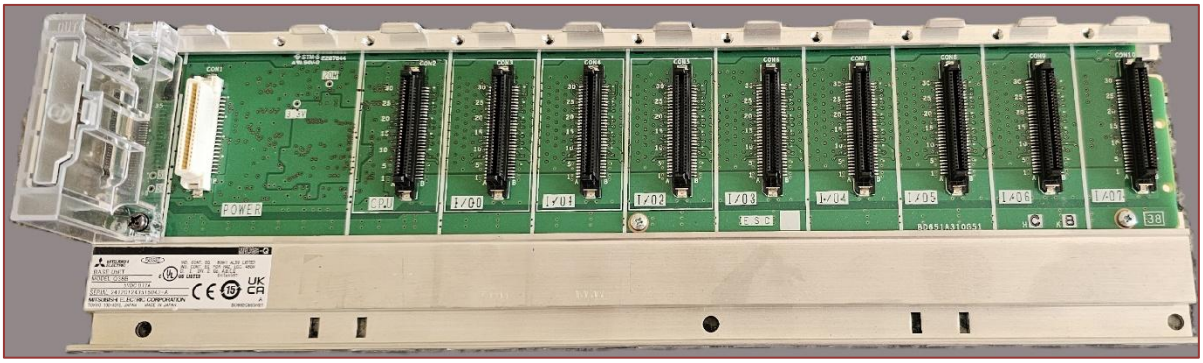


Fonte: Petruzella, 2014.

Para ministra a disciplina foram utilizados 3 CLPs modulares da Série Q da fabricante Mitsubishi. Contendo os seguintes componentes:

- Unidade base: codificação Q38B com 8 slots, ilustrado na Figura 3;
- Fonte de alimentação: codificação Q61P 100 – 240 Vac
- CPU: codificação Q03UDV
- Módulo de entrada: codificação QX41
- Módulo de saída: codificação QY41P

Figura 6.3 - Unidade base com 8 *slots*



Fonte: Os autores, 2025.

A unidade base Q38B é caracterizada por três conexões iniciais sequenciais: a de extensão, alimentação e CPU. Após essa etapa, disponibiliza oito slots codificados como I/00, I/01, I/02, I/03, I/04, I/05, I/06 e I/07, destinados à conexão de CPUs, módulos de comunicação e módulos de entrada/saída, conforme destacado na Figura 6.3.

Quanto à fonte de alimentação Q61P, ela opera com uma faixa de tensão de entrada de 100 a 240 VAC, uma frequência de 50/60 Hz e fornece uma tensão de saída de 5 VDC. A correta conexão desta fonte à rede elétrica de 127/220 VAC é apresentada de maneira detalhada na Figura 6.4, fornecendo uma referência visual para a instalação adequada.

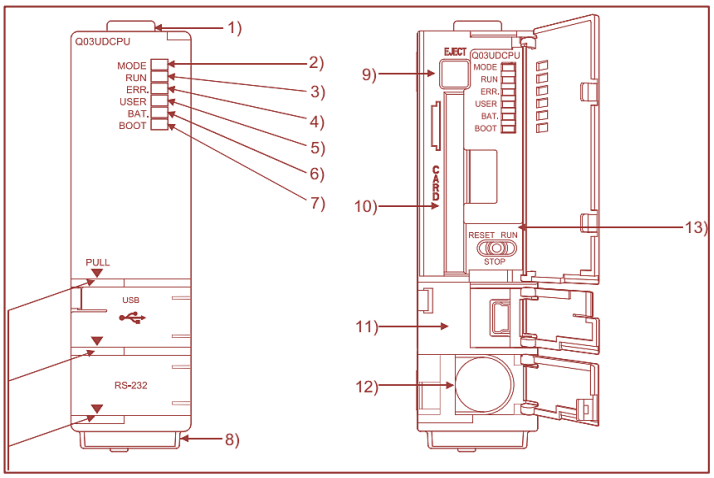
Figura 6.4 - Fonte de alimentação Q61P

Fonte: Os autores, 2025.

A CPU Q03UDV apresenta capacidade de programação de 30.000 passos e capacidade de gerenciamento de 4096/8192 pontos de entradas/saídas. Além disso esta

CPU possui período do ciclo de programa de 20 ns/*log. Instruction*. Dentro da CPU existe uma bateria (item 12 na Figura 6.5) com vida útil de 43.800 horas (5 anos).

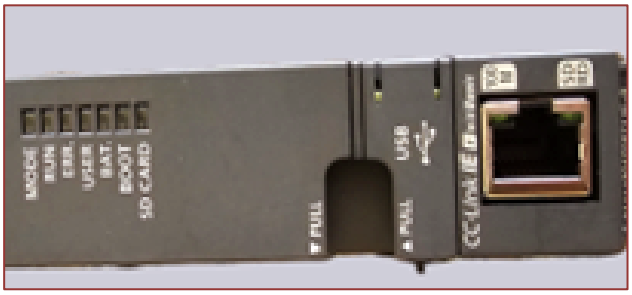
Figura 6.5 - Unidade central de processamento Q03UD



Fonte: Mitsubishi Electric Corporation, 2004.

Na Figura 6.5 são ilustradas as principais partes de um módulo CPU. No item 1 é identificado o gancho utilizado para fixar o módulo à unidade base. No item 2 identifica-se o Led responsável por indicar o modo da CPU, ou seja, ligado ou biscando (a função de ativação/desativação forçada de entrada/saída externa está sendo executada). Por sua vez, o led RUN (item 3) pode estar em três estados: ligado (quando o interruptor RUN/STOP/RESET (item 13) estiver em RUN), desligado (quando o interruptor RUN/STOP/RETET estiver em STOP) e piscando (quando o interruptor RUN/STOP/RETET mudar de STOP para RUN). O item 4 identifica o led ERR responsável por indicar a detecção de um erro que não parou a operação (ligado), estado normal (desligado) e piscando quando identificado um erro que interrompe a operação. O led USER (item 5) ligado quando annunciator (F) ligado e desligado em estado normal. Já o led BAT no item 6, quando ligado, indica erro na bateria devido à redução nas tensões da bateria do módulo CPU ou cartão de memória, caso contrário, estado normal. A Figura 6.6 ilustra uma foto do módulo CPU com os leds indicativos.

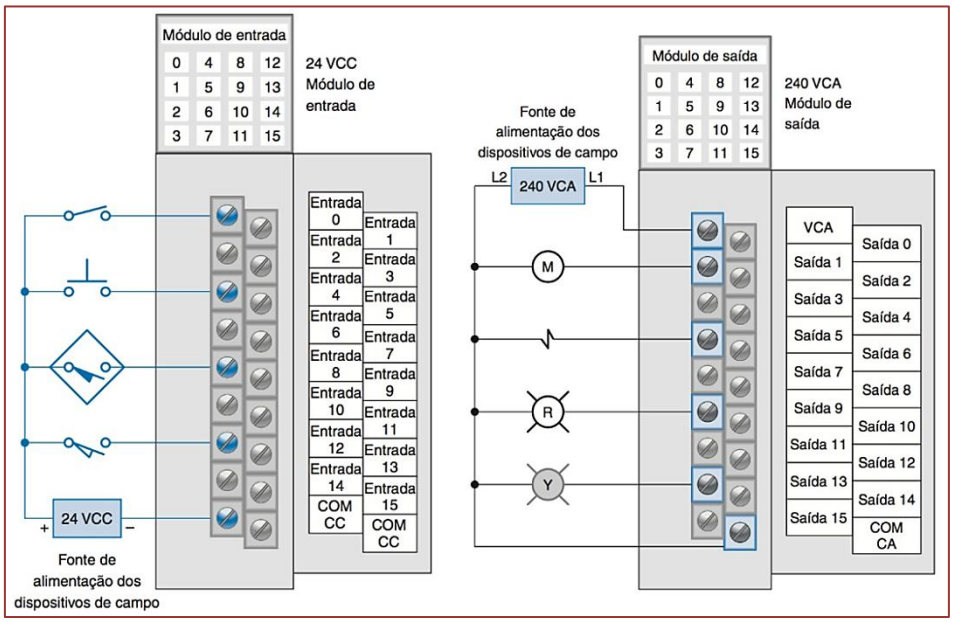
Figura 6.6 - Unidade central de processamento Q03UDV utilizadas nas aulas



Fonte: Autor, 2023.

Os módulos de entradas/saídas desempenham um papel fundamental na automação industrial, conectando dispositivos de campo ao controlador. Os módulos de entrada recebem sinais de sensores, botões e chaves-limite, fornecendo informações em tempo real. Já os módulos de saída enviam sinais para dispositivos como sinaleiras, válvulas solenoides e motores, traduzindo os comandos do controlador em ações físicas (Figura 6.7). Essa interação eficiente é essencial para otimizar processos, garantindo uma resposta ágil e precisa do sistema a mudanças nas condições de operação. Em resumo, esses módulos são cruciais para a eficiência, confiabilidade e adaptação do sistema.

Figura 6.7 - Ilustração dos módulos de entrada e saída



Fonte: Petruzella, 2014.

Os módulos de entradas e saídas podem ser classificados quando ao sentido do fluxo de corrente convencional (do menor potencial ao maior potencial) entre dois dispositivos como *sinking* (dreno) e *sourcing* (fonte).

Considere dois dispositivos genéricos: dispositivo 1 e dispositivo 2. Se o sentido do fluxo de corrente é do dispositivo 1 ao dispositivo 2, então denomina-se ao dispositivo 1 como do tipo *sourcing* e o dispositivo 2 como do tipo *sinking*. Por outro lado, se o sentido do fluxo de corrente for do dispositivo 2 ao dispositivo 1, chamar-se-á o dispositivo 2 de *sourcing* e o dispositivo 1 de *sinking*. A Tabela 6.1 resume esta classificação.

Tabela 6.1 - Classificação de dispositivos em relação ao sentido do fluxo de corrente

Sentido do Fluxo de Corrente	Dispositivo 1	Dispositivo 2
Do **Dispositivo 1** ao **Dispositivo 2**	*Sourcing* (**+ comum**)	*Sinking* (**- comum**)
Do **Dispositivo 2** ao **Dispositivo 1**	*Sinking* (**- comum**)	*Sourcing* (**+ comum**)

Fonte: Os autores, 2025.

Para um módulo funcionar como *sourcing*, o comum deverá estar conectado a um potencial positivo, de outra forma, para funcionar como *sinking*, o comum deverá estar conectado ao negativo do módulo.

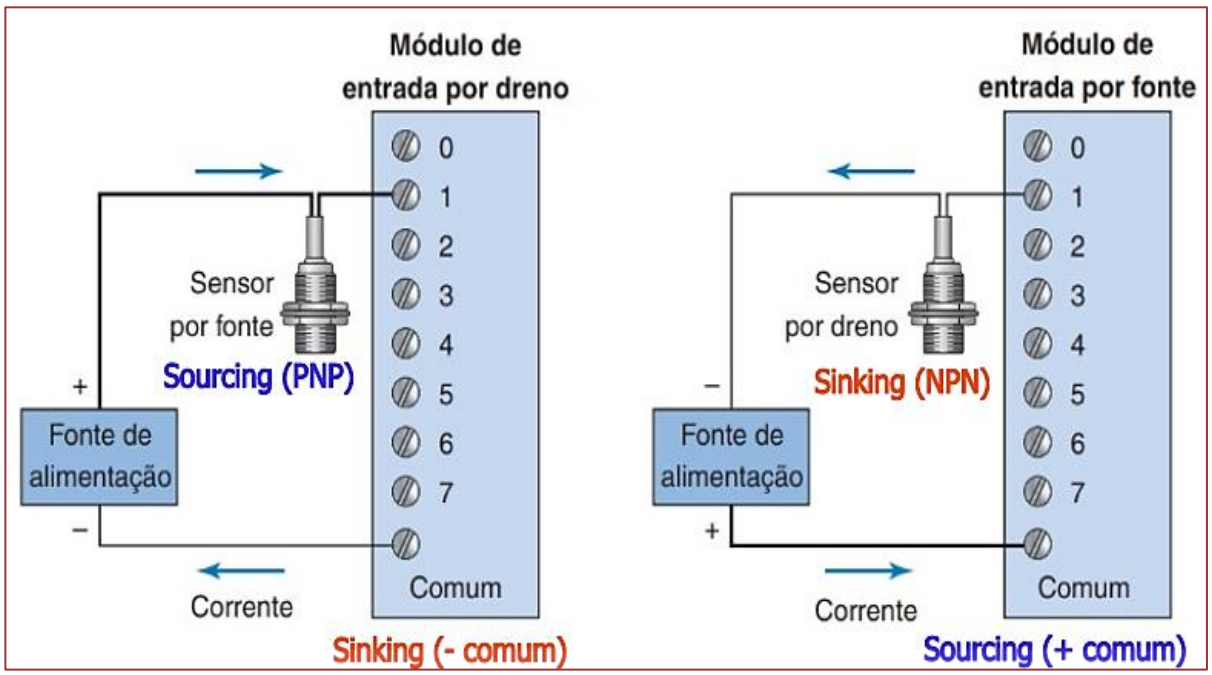
Considerando tais fatos, se um módulo de entrada é do tipo *sourcing* então o sensor conectado neste módulo é do tipo NPN. Caso contrário, módulo de entrada do tipo *sinking*, então o sensor conectado neste módulo será PNP. A Tabela 6.2 e Figura 6.8 resumem e ilustram esta classificação.

Tabela 6.2 - Classificação de módulos e sensores

Sentido do Fluxo de Corrente	Módulo	Sensor
Do **Módulo** ao **Sensor**	*Sourcing* (**+ comum**)	*Sinking* (**NPN**)
Do **Sensor** ao **Módulo**	*Sinking* (**- comum**)	*Sourcing* (**PNP**)

Fonte: Os autores, 2025.

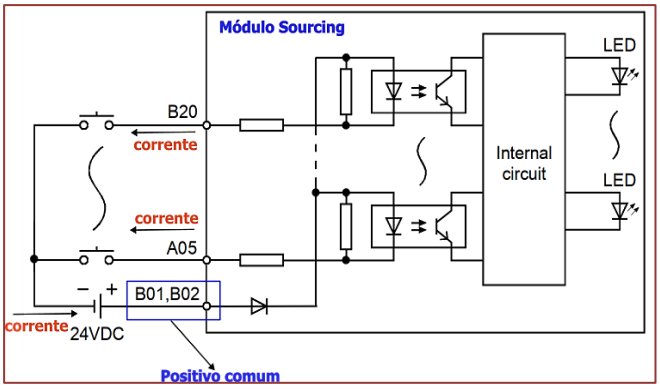
Figura 6.8 - Ilustração da classificação de módulos e sensores



Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

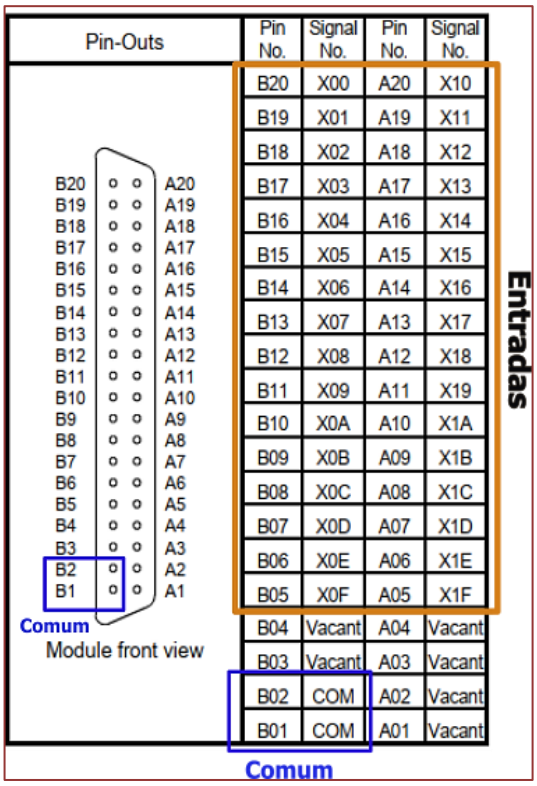
O módulo QX41 possui 32 pontos, método de isolamento fotoacoplador, tensão de entrada de 24 VDC e corrente de entrada nominal de 4 mA. O circuito de conexão externa do módulo é ilustrado na Figura 6.9, que apresenta um módulo positivo comum, portanto do tipo *sourcing*. A Figura 6.10 apresenta a pinagem do módulo QX41.

Figura 6.9 - Circuito de conexão externa QX41



Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

Figura 6.10 - Pinagem do módulo QX41

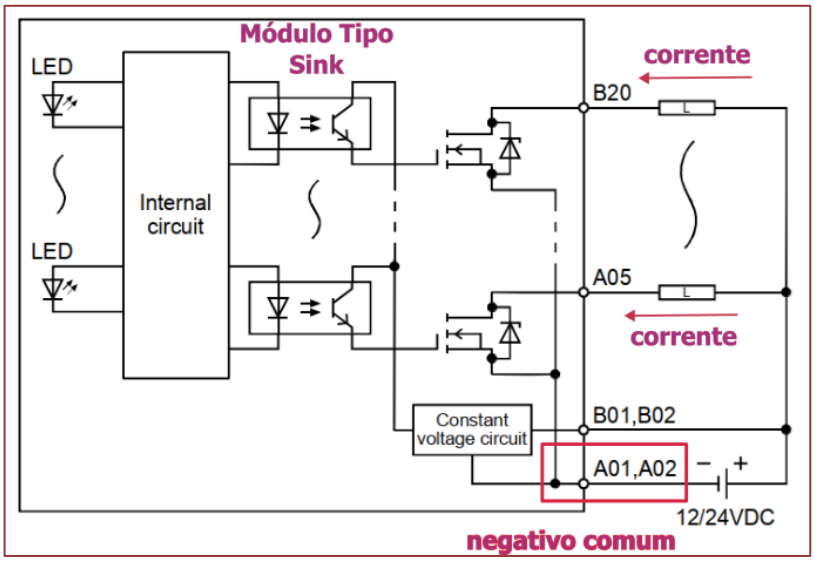


Fonte: Mitsubishi Electric Corporation, 2004).

O módulo de saída digital QY41P apresenta 32 pontos de saída e utiliza o método de isolamento fotoacoplador para assegurar confiabilidade operacional. Com uma faixa de tensão nominal de 12 a 24 V DC, o dispositivo suporta uma máxima corrente de carga de 0.1 A por ponto e 2 A por comum, proporcionando versatilidade em aplicações industriais. O tempo de resposta é de 1 ms, garantindo uma atuação rápida e precisa. O módulo incorpora um supressor de surto com diodo Zener para proteção contra picos de tensão. Possui resistência de isolamento de 10 MΩ e utiliza o conector A6CON2. Como

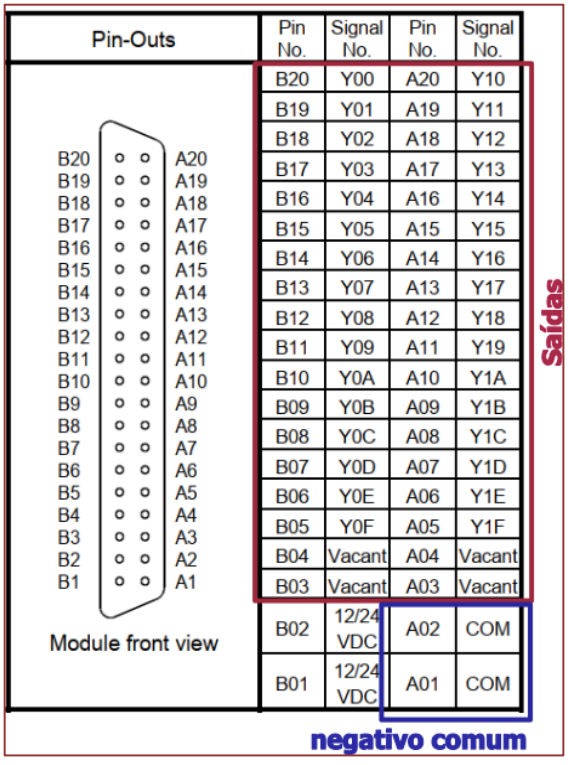
pode ser observado na Figura 6.11, o módulo QY41P possui negativo comum, portanto é um módulo do tipo sink. A Figura 6.12 apresenta a pinagem módulo QY41P, e a Figura 6.13 traz uma foto do módulo de entrada digital QX41 e módulo de saída digital QY41P.

Figura 6.11 - Circuito de conexão externa QY41P



Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

Figura 6.12 - Pinagem módulo QY41P



Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

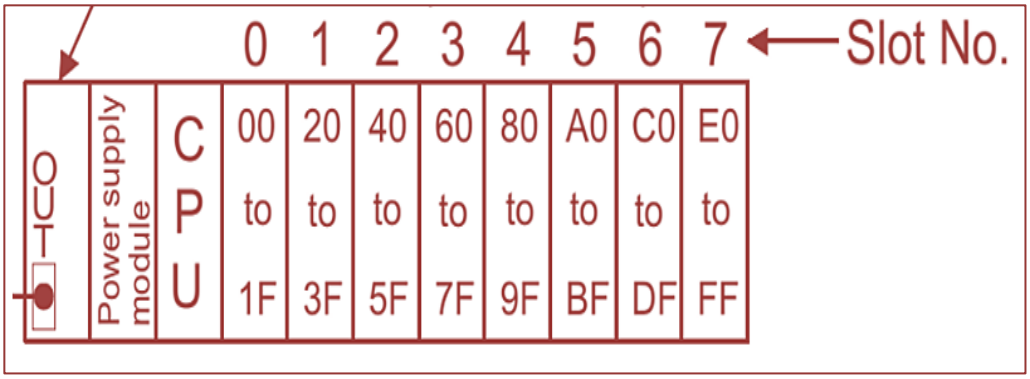
Figura 6.13 - Módulo de entrada digital QX41 e módulo de saída digital QY41P

Fonte: Autores, 2025.

Como pode ser observado na Figura 6.11, o módulo QY41P tem o menor potencial da fonte de 12/24 VDC conectado nos terminais comuns A01 e A02, caracterizando assim um módulo negativo comum, ou seja, um módulo do tipo *sink*.

A codificação das entradas e saídas dos módulos na unidade base obedece a uma codificação em representação hexadecimal, como mostrada na Figura 6.14.

Figura 6.14 - Codificação das entradas/saídas em hexadecimal

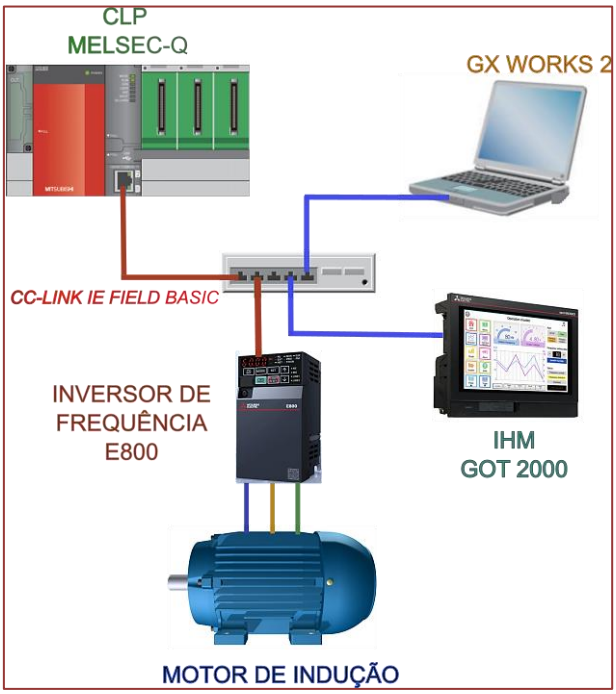


Fonte: Adaptado de Mitsubishi Electric Corporation, 2004.

6.2.2. EXEMPLO DE PROJETO PRÁTICO DA DISCIPLINA

O sistema apresentado na Figura 6.15 ilustra uma arquitetura de controle industrial compacta, composta por um CLP da série MELSEC-Q, uma Interface Homem-Máquina (IHM) da série GOT 2000, um Inversor de Frequência modelo FR-E800 e um motor de indução. A interconexão desses dispositivos é realizada por meio de um hub, utilizando o software de programação GX Works 2 para o desenvolvimento e comissionamento do sistema.

Figura 6.14 - Arquitetura de sistema de automação de acionamento de um motor com via inversor de frequência



Fonte: Os autores, 2025.

A integração entre o CLP, que atua como mestre da rede, e o inversor de frequência é estabelecida através do protocolo de comunicação CC-Link IE Field Basic. Este é um protocolo de rede baseado em Ethernet de uso geral que permite a coexistência com outras comunicações baseadas em TCP/IP, tornando a arquitetura de rede flexível e expansível.

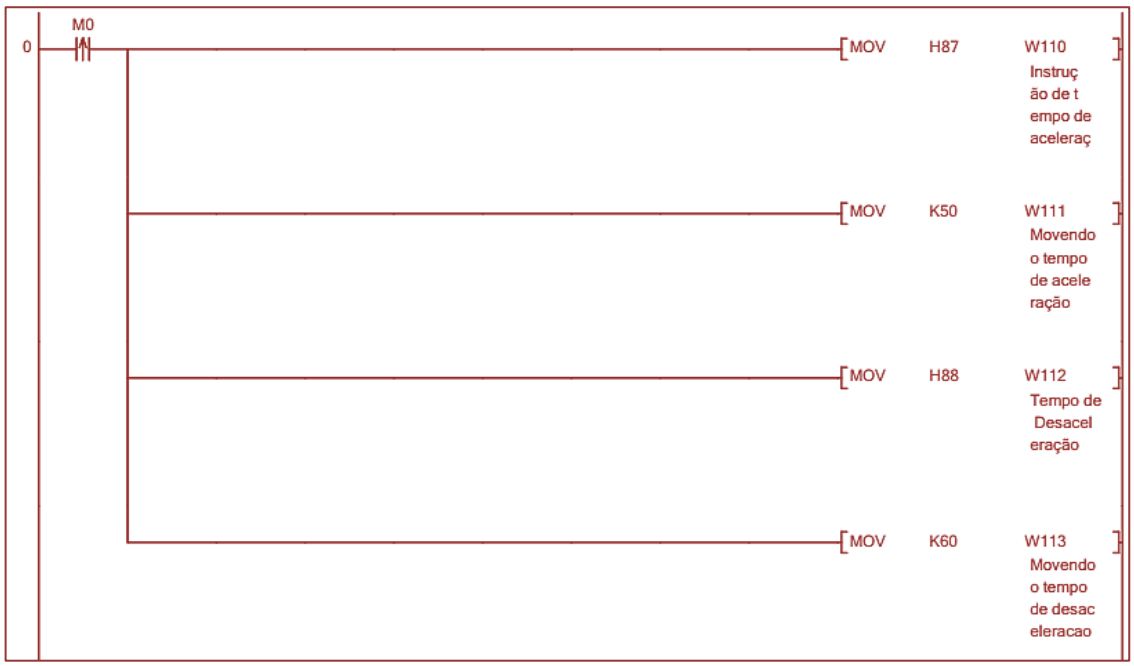
O objetivo central desta aplicação foi realizar a integração completa entre o CLP e o inversor de frequência para obter o controle e o monitoramento de um motor de indução. A prática visou capacitar o CLP a realizar as seguintes operações: (a) enviar comandos de partida em avanço e reverso, e principalmente, inserir dinamicamente valores de setpoint de velocidade (frequência) para o motor; (b) receber do inversor informações em tempo real sobre o estado do motor, como a velocidade atual, a corrente consumida e a tensão de saída.

O desenvolvimento da solução foi estruturado em três etapas fundamentais, seguindo uma sequência lógica desde a configuração do hardware de campo até a codificação da lógica de controle.

1. Parametrização do Inversor de Frequência: Inicialmente, foi realizada a parametrização do inversor FR-E800 envolvendo a configuração de parâmetros essenciais para a comunicação, como a seleção do protocolo CC-Link IE Field Basic (e.g., Pr. 1427 = 61450), a definição de um endereço IP para o inversor (Pr. 1434 a Pr. 1437) e a especificação do endereço IP do CLP mestre autorizado a enviar comandos (Pr. 1449 a Pr. 1452). Após a parametrização, um reinício do equipamento foi necessário para aplicar as novas configurações.

2. Configuração no Ambiente de Programação (GX Works 2): Com o inversor devidamente preparado, o passo seguinte foi a configuração do CLP no software GX Works 2. Primeiramente, foi necessário registrar o profile do inversor no ambiente de desenvolvimento, um arquivo que descreve o dispositivo para que o software o reconheça na rede. Em seguida, nas configurações da porta Ethernet do CLP, a função CC-Link IEF Basic foi habilitada. A etapa final desta fase foi a configuração da rede, onde o inversor foi detectado e adicionado como uma estação escrava. Realizou-se então o mapeamento dos registradores remotos de troca de dados — RX/RY (bits de status/comando) e RWr/RWw (palavras de leitura/escrita) — para endereços específicos na memória do CLP.
3. Desenvolvimento do Programa em Ladder: Concluída a configuração da comunicação, a lógica de controle foi implementada em linguagem Ladder. Utilizando os endereços de memória previamente mapeados, o programa foi desenvolvido para escrever valores nos registradores de comando (RWw), como o RefVeloc_INV_n para o setpoint de frequência, e para acionar bits de comando (RY), como o STF_INV_n para partida em avanço. De forma análoga, a lógica de monitoramento foi implementada para ler os valores dos registradores de status (RWr), como VelAtual INV n (velocidade atual) e CorrenteAtual_INV_n (corrente atual), permitindo que o CLP tomasse decisões ou apenas exibisse os dados na IHM.

Figura 6.15 - Primeira parte do código de integração CLP ao inversor



Fonte: Os autores, 2025.

O segmento de código em Ladder ilustrado na Figura 6.15 é executado quando o contato do relé interno M0 é ativado. Sua função é preparar os valores para a configuração dos tempos de aceleração e desaceleração de um inversor de frequência. As duas primeiras instruções MOV (Move) trabalham em conjunto: a primeira move o valor hexadecimal H87 para o registrador W110, que corresponde ao código de instrução para

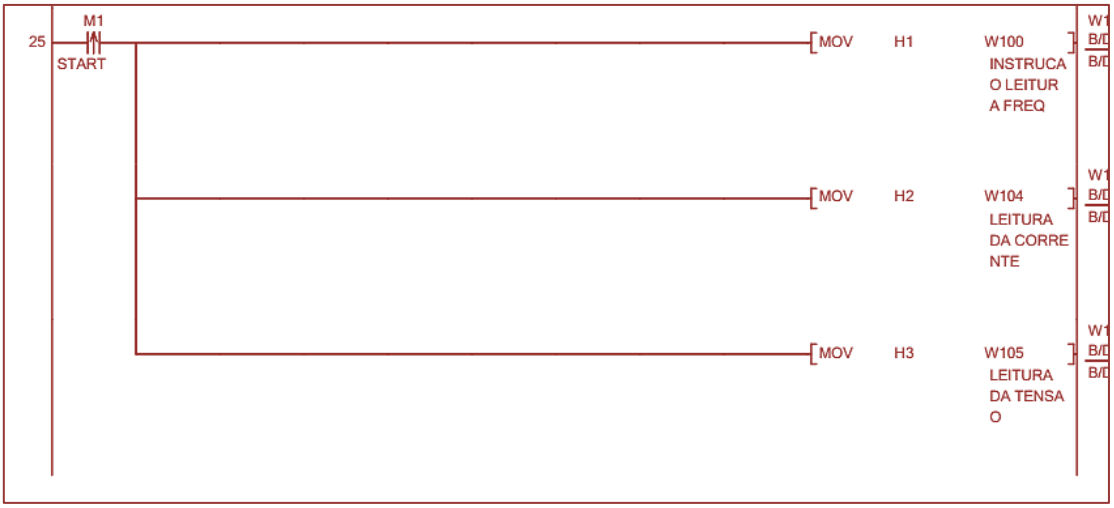
escrita no parâmetro de tempo de aceleração (Parâmetro 7). A segunda instrução move a constante decimal K50 para o registrador W111, definindo o valor do tempo (e.g., 5,0 segundos). De forma análoga, as duas instruções seguintes preparam o tempo de desaceleração: o valor H88 (código de instrução para o Parâmetro 8) é movido para W112, e a constante K60 (e.g., 6,0 segundos) é movida para W113. Essencialmente, este trecho de código carrega os códigos dos parâmetros e os respectivos valores desejados em registradores de preparação, que serão posteriormente enviados ao inversor através de uma rotina de comunicação.

Figura 6.16. - Segunda parte do código de integração CLP ao inversor

Fonte: Os autores, 2025.

O trecho do programa em Ladder ilustrado na Figura 6.16 implementa a lógica fundamental de partida, parada e ajuste de velocidade de um motor controlado por inversor. A lógica de partida é iniciada pelo contato M1 (START), que utiliza uma instrução SET para ligar e travar a saída Y1000, correspondente ao comando de avanço do motor. Uma vez que Y1000 está ativa, uma instrução MOV é executada para mover o valor constante K6000 (representando um setpoint de 60,00 Hz) para o registrador W101. O envio deste valor ao inversor, bem como a execução de outras instruções, é garantido pelo pulso de clock SM412 na primeira linha, que aciona periodicamente as saídas Y100F e Y100E (solicitação de execução de instrução e comando de escrita de frequência, respectivamente). Por fim, a parada do motor é realizada ao acionar o contato M2 (STOP), que executa uma instrução RST (Reset) na saída Y1000, desativando o comando de avanço.

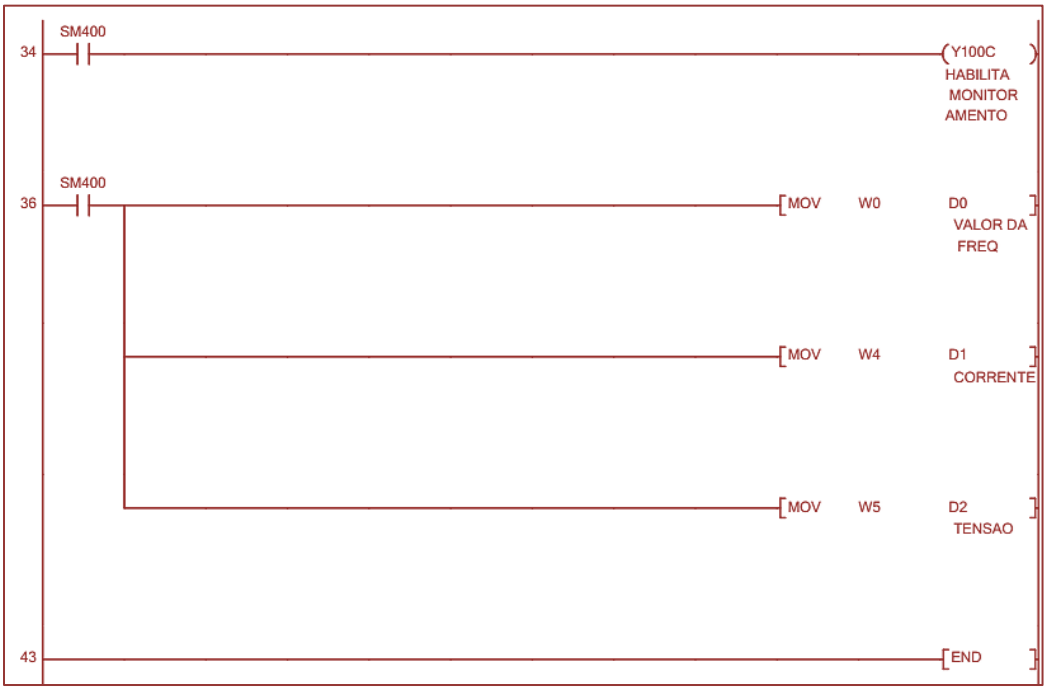
Figura 7.17 - Terceira parte do código de integração CLP ao inversor



Fonte: Os autores, 2025.

Na Figura 6.17 ilustra-se o acionamento simultaneamente ao comando de partida do motor, através do contato M1 (START), e sua função é configurar as solicitações de monitoramento de variáveis do inversor. Quando M1 é ativado, três instruções MOV são executadas em paralelo. A primeira move o valor hexadecimal H1 para o registrador W100, que corresponde ao código de instrução para a leitura da frequência de saída. A segunda move o valor H2 para W104, configurando a solicitação de leitura da corrente de saída. A terceira move H3 para W105, preparando a solicitação para leitura da tensão de saída. Esses registradores (W100, W104, W105) são parte da área de dados de escrita (RWw) na comunicação com o inversor. Uma vez que esses códigos de monitoramento são enviados, o inversor responderá preenchendo os registradores de leitura (RWr) correspondentes com os valores atuais de frequência, corrente e tensão, permitindo que o CLP monitore o estado do motor em tempo real.

Figura 6.18 - Quarta parte do código de integração CLP ao inversor



Fonte: Os autores, 2025.

Na Figura 6.18 ilustra-se o trecho final do código que é executado continuamente a cada ciclo de varredura do CLP, pois é condicionado pelo contato especial SM400, que permanece sempre ativo enquanto o CLP está em modo RUN. A primeira linha do código mantém a saída Y100C constantemente energizada, que está mapeada para o "Comando de Monitoramento" (RYC) do inversor. Manter este comando ativo instrui o inversor a atualizar continuamente os valores de suas variáveis de leitura com base nos códigos de monitoramento solicitados anteriormente. A segunda parte do código, também sempre ativa, utiliza três instruções MOV para copiar os dados recebidos do inversor (que chegam nos registradores de leitura RWr, mapeados para a área W do CLP) para registradores de dados internos (D). Especificamente, o valor do registrador W0 (contendo a frequência atual) é movido para D0, o valor de W4 (corrente) é copiado para D1, e o de W5 (tensão) é copiado para D2. Em suma, este bloco finaliza o ciclo de monitoramento, garantindo que o inversor envie os dados e que o CLP os armazene em uma área de trabalho para uso em lógicas de controle ou para exibição em uma IHM.

6.3. PROCESSO AVALIATIVO

Nesta disciplina, a avaliação teve como foco a análise qualitativa do progresso na assimilação de novos conceitos e no aprimoramento de habilidades por parte dos alunos, ao longo do ciclo de criação de soluções utilizando o *Design Thinking*. Além disso, incluiu-se a avaliação da execução de planos de solução fundamentados na abordagem de aprendizagem baseada em projeto.

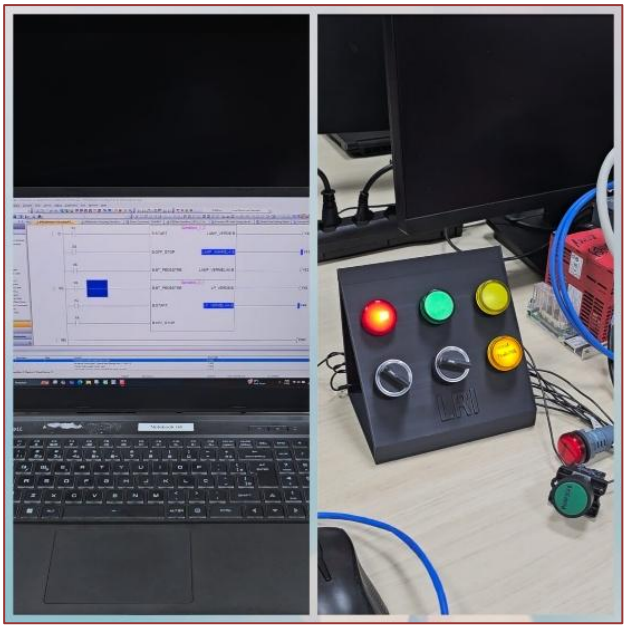
Ao longo da disciplina, três grupos de cinco alunos foram expostos a situações-problema (projetos parciais) de programação em CLP que tinham como objetivo adquirir conhecimentos teóricos e habilidades práticas de programação em CLP. Estes alunos

foram avaliados quanto ao engajamento; pesquisa e ideação; desenvolvimento e comunicação. Cada um destes indicadores foi qualificado como: superou as expectativas; atendeu às expectativas; precisa melhorar e inadequado.

Os seguintes projetos parciais foram desenvolvidos durante a disciplina:

- Projeto parcial 1: aquisição de habilidades de programação nas linguagens *ladder*, diagrama de blocos e texto estruturado utilizando o software GXWorks 2, a Figura 6.19 retrata este momento.

Figura 6.19. Aluno programando um CLP utilizando bloco de função (Projeto Parcial 1)



Fonte: Os autores, 2023.

- Projeto parcial 2: simulação de projetos de automação por meio da integração do software de simulação FactoryIO e o GXWorks 2, e componentes elétricos. A Figura 6.20 retrata este momento.

Figura 6.20. Alunos automatizando um ambiente simulado no software FactoryIO integrado ao GXWorks 2 (Projeto Parcial 2)

Fonte: Os autores, 2025.

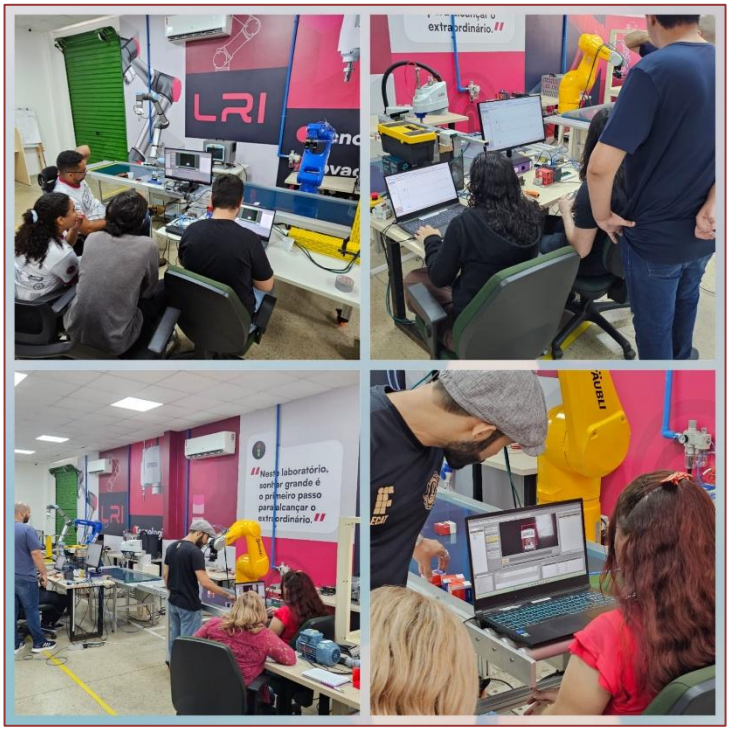
- Projeto parcial 3: integração do CLP da Mitsubishi Série Q ao sistema de visão Cognex. A Figura 5.21 retrata este momento.

Figura 6.21. Alunos integrando sistema de visão Cognex ao CLP Mitsubishi utilizandos os softwares In-Sight Explore e GXWorks 2 (Projeto Parcial 3)

Fonte: Os autores, 2025.

- Projeto parcial 4: integração do CLP da Mitsubishi Série Q, inversor de frequência, sensor de visão Cognex e esteira. A Figura 6.22 retrata este momento.

Figura 6.22. Integração CLP, inversor de frequência, câmera Cognex e esteira (Projeto Parcial 4)



Fonte: Os autores, 2025.

6.4. RESULTADOS DA DISCIPLINA

Os resultados alcançados pela disciplina de Programação de CLP foram:

- Identificar e Descrever Componentes de Hardware de CLP: Diferenciar e explicar a função dos componentes fundamentais de um sistema de CLP modular, incluindo rack, fonte, CPU e módulos de entrada/saída, sendo capaz de especificar hardware para aplicações simples.
- Dominar o Ambiente de Programação: Utilizar com proficiência o software GX Works 2 para criar, editar, compilar, depurar e monitorar programas de automação, bem como realizar o upload e download de projetos para o controlador.
- Programar em Linguagem Ladder: Desenvolver lógicas de controle utilizando a linguagem Ladder para resolver problemas industriais clássicos, aplicando corretamente conceitos de contatos, bobinas, selo, intertravamento, temporizadores e contadores.
- Aplicar Múltiplas Linguagens de Programação: Projetar soluções de automação de forma estruturada e eficiente, escolhendo e aplicando a linguagem de programação

mais adequada para cada tarefa, seja ela Ladder, Diagrama de Blocos de Funções (FDB) ou Texto Estruturado (ST).

- Utilizar Simulação para Validação de Projetos: Criar ambientes virtuais de máquinas e processos no software Factory IO, integrando-os ao CLP para testar, validar e otimizar a lógica de controle em um ambiente seguro antes da implementação em hardware real.
- Integrar e Controlar Acionamentos Elétricos: Configurar e programar a comunicação entre o CLP e inversores de frequência para controlar a velocidade, a aceleração/desaceleração e o sentido de rotação de motores de indução em aplicações como esteiras transportadoras.
- Configurar Redes de Comunicação Industrial: Estabelecer a comunicação entre dispositivos de automação utilizando protocolos de rede CC-Link IE Field Basic.
- Integrar Sistemas de Visão Computacional: Implementar soluções que utilizam sensores de visão (COGNEX) conectados ao CLP para realizar tarefas de inspeção, contagem, classificação de peças ou controle de qualidade em uma linha de produção.
- Desenvolver Projetos de Automação Completos: Conceber, planejar e executar um projeto de automação de pequeno porte do início ao fim, integrando múltiplos componentes (CLP, inversor, motor, sensores) e desenvolvendo uma solução funcional para um problema prático da indústria.

A disciplina de Programação em CLP proporcionou uma experiência de aprendizado abrangente e altamente eficaz para os alunos. A estrutura do curso, que incluiu uma combinação equilibrada de teoria e prática, permitiu que os estudantes desenvolvessem uma compreensão profunda dos conceitos fundamentais da programação de Controladores Lógicos Programáveis. As atividades práticas, desde a programação em linguagem ladder até a integração de tecnologias avançadas como visão computacional e robótica industrial, desempenharam um papel crucial no fortalecimento das habilidades técnicas dos alunos. A utilização de metodologias ativas de aprendizagem proporcionou uma dinâmica envolvente, incentivando a participação ativa dos alunos e promovendo uma aplicação prática dos conhecimentos adquiridos.

Além disso, a diversidade de tópicos, que abrangeu desde programação de CLPs até a programação de robôs industriais e sistemas de visão, refletiu a natureza abrangente da automação industrial. A habilidade dos alunos em integrar esses conceitos e tecnologias diversos demonstra não apenas uma compreensão pontual, mas uma visão integrada e holística da automação. A incorporação de habilidades de aprendizagem ativa, juntamente com a aplicação prática dos conhecimentos, solidificou a preparação dos alunos para enfrentar desafios complexos e se destacar em ambientes industriais modernos.

REFERÊNCIAS

[1] DA SILVA, Edilson Alfredo. **Introdução às linguagens de programação para CLP**. Blucher, 2016.

[2] MALEKAR, Avinash. Learn everything about PLC programming: Practical lessons on Allen-Bradley, Siemens, and Mitsubishi PLC with real world examples**. Independently published**, 2021.

[3] Mitsubishi Electric Corporation. **QCPU User's Manual** (Hardware Design, Maintenance and Inspection). Manual number SH(NA)-080483ENG-AZ(2309)MEE, 2004.

[4] Mitsubishi Electric Corporation. I/O Module Type Building Block – **User's Manual. Manual number SH(NA)-080042-T(0810) MEE**, 2000.

[5] PETRUZELLA, Frank D. **Controladores lógicos programáveis**. AMGH Editora, 2014.

[6] PRUDENTE, Francesco. Automação Industrial PLC - Teoria e Aplicações - Curso Básico. **Grupo Gen-LTC**, 2015.

[7] QUEIROZ-NETO, J. P., SALES, D. C., PINHEIRO, H. S., NETO, B. O. Using modern pedagogical tools to improve learning in technological contents. 2015 **IEEE Frontiers in Education Conference (FIE)**, El Paso, TX, 2015, pp. 1-8, doi: 10.1109/FIE.2015.7344383.

[8] WHITE, M. T. Mastering PLC Programming: **The software engineering survival guide to automation programming**. Packt Publishing, 2023.

CAPÍTULO

07 Programação em *Interface* Homem Máquina - IHM

Vanderson de Lima Reis[1], Hugo Alves Velozo[2]

7.1. CARACTERIZAÇÃO DA DISCIPLINA

Na disciplina de Programação de IHM, antes de começar a parte mais prática e de programação, é feita uma introdução dos conceitos, definições e fluxos de projeto que podem ser utilizados na Programação das IHMs. Contudo é na execução das práticas e dos projetos que muitos desses conceitos vão sendo mais bem entendidos e absorvidos pelos alunos, lhe dando um sentimento maior das opções de como um determinado projeto pode ser feito, que em língua inglesa usamos o termo *feeling* para tal situação.

Nesta disciplina, como nas demais, foi feita com a integração das metodologias de Aprendizagem Baseada em Projetos (PBL) e *Design Thinking* (DT), criando métricas ou indicadores para verificar como foi o rendimento desses alunos com essa integração.

7.1.1. EMENTA

Conceitos, funcionamento e histórico das IHMs. Apresentação geral do *hardware* e *Softwares* do fabricante, e seus principais manuais de criação de projeto no *software* do fabricante e suas configurações básicas, como: edição, compilação, simulação, programação e monitoramento da IHM. Simulação da IHM em conjunto com a simulação do CLP (esse último usando programação em Ladder e em Texto Estruturado). Programar um projeto na IHM real, comunicação, manuseio e ensaios. Construção das Telas da IHM. Inserindo e configurando elementos de Imagens, Lâmpadas e Botões *etc*. Inserindo e configurando elementos de display numérico e texto, data e hora, receita, alarmes e segurança. Animação e Movimento de Objetos na IHM. Comunicação da IHM com CLP ou outros periféricos via padrão Ethernet dentre outros protocolos de comunicação industrial.

7.1.2. OBJETIVOS

Desenvolver no discente os conceitos básicos do que é uma Interface Homem Máquina (IHM), suas aplicações e limitações.

Desenvolver sua habilidade para fazer projetos em IHMs, mostrando como o projeto da IHM pode estar vinculado diretamente com um Controlador Lógico Programável (CLP) dentre outras opções.

[1] Doutor em Engenharia Elétrica. Professor da disciplina de Programação em IHM. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). E-mail: *vanderson.reis@ifam.edu.br.*
[2] Mestre em Automação e Controle. Professor da disciplina de Programação em CLP. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). E-mail: *hugo.velozo@ifam.edu.br.*

7.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 20h

Carga Horária Prática: 40h

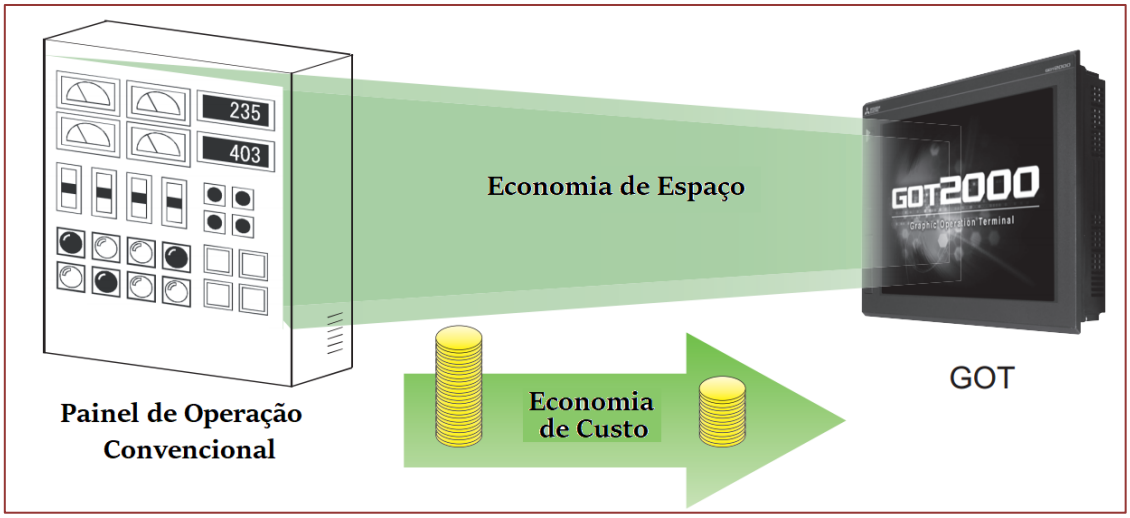
Total: 60h

7.2. CONTEÚDOS

7.2.1. O QUE É UMA IHM

A rigor poderíamos dizer que qualquer mecanismo que faz uma integração entre um operador ("Homem") com uma máquina é uma Interface Homem Máquina, como exemplo um painel com botões, chaves, sinaleiras, *displays* e *etc*. Contudo no ambiente industrial contemporâneo uma IHM é muitas vezes conhecida como um *display* gráfico com *touch*, capaz de ser programado para criar de forma gráfica todos os dispositivos contidos em um painel de operação convencional e outros mais, como gráficos e animações. Talvez o termo IHM não seja o mais correto para definir essas IHMs modernas, alguns fabricantes costumam a dar seus próprios nomes para suas IHMs (mas claro que isso é jargão de empresa), como exemplo a MITSUBISHI usa o termo GOT para suas IHMs. GOT é um acrônimo para *Graphic Operation Terminal*, que fazendo uma tradução para língua portuguesa, ficaria algo como Terminal de Operação Gráfico (TOG). Na Figura 7.1 é possível comparar visualmente um painel de operação convencional com uma IHM da MITSUBISHI, uma GOT da sua linha GOT2000.

Figura 7.1 – Comparativo entre uma IHM antiga e uma moderna



Fonte: Editado de MITSUBISHI 04, 2018.

Na própria Figura 7.1 já é possível observar alguns benefícios ao usar uma IHM moderna, no caso uma GOT. Contudo vamos listar em seguida algumas dessas vantagens (MITSUBISHI 04, 2018):

1) Redução do painel de operação: O GOT substitui chaves e lâmpadas físicas por gráficas, economizando pontos de E/S no PLC e reduzindo o tamanho do painel de operação.
2) Redução dos custos de fiação: A interface gráfica do usuário do GOT elimina o tempo e o custo de cabeamento de componentes dentro da operação painel.
3) Padronização de painéis de operação: Os GOTs são flexíveis, permitindo construir ou modificar projetos sem alterar o programa PLC ou a fiação do sistema, o que facilita a padronização dos painéis de operação.
4) IHM (Interface Homem-Máquina) de valor agregado: GOT também pode exibir gráficos, textos e alarmes, além de interruptores e lâmpadas, contribuindo com valor além de todo o equipamento.

Como nesse curso as práticas e os projetos foram realizados utilizando a IHM da MITSUBISHI, vamos focar nossas explicações para IHM desse fabricante, ficando claro que um caminho muito parecido poderia ser feito para outros fabricantes.

Embora seja possível utilizar uma IHM sem o uso de um CLP (Controlador Lógico Programável), na grande maioria dos casos as IHM trabalham em conjunto com um CLP. Vale destacar que muitos fabricantes de IHMs são também fabricantes de CLPs, de tal forma que a integração entre a IHM e o CLP do mesmo fabricante costuma ser mais simples, do que se os fabricantes forem diferentes, neste nosso curso usamos tanto a IHM quando o CLP do mesmo fabricante, a MITSUBISHI, o curso como um todo foi idealizado de forma que a disciplina de IHM viesse *a posteriori* ao curso de CLP. Por tanto os alunos já sabem programação LADDER e Texto Estruturado.

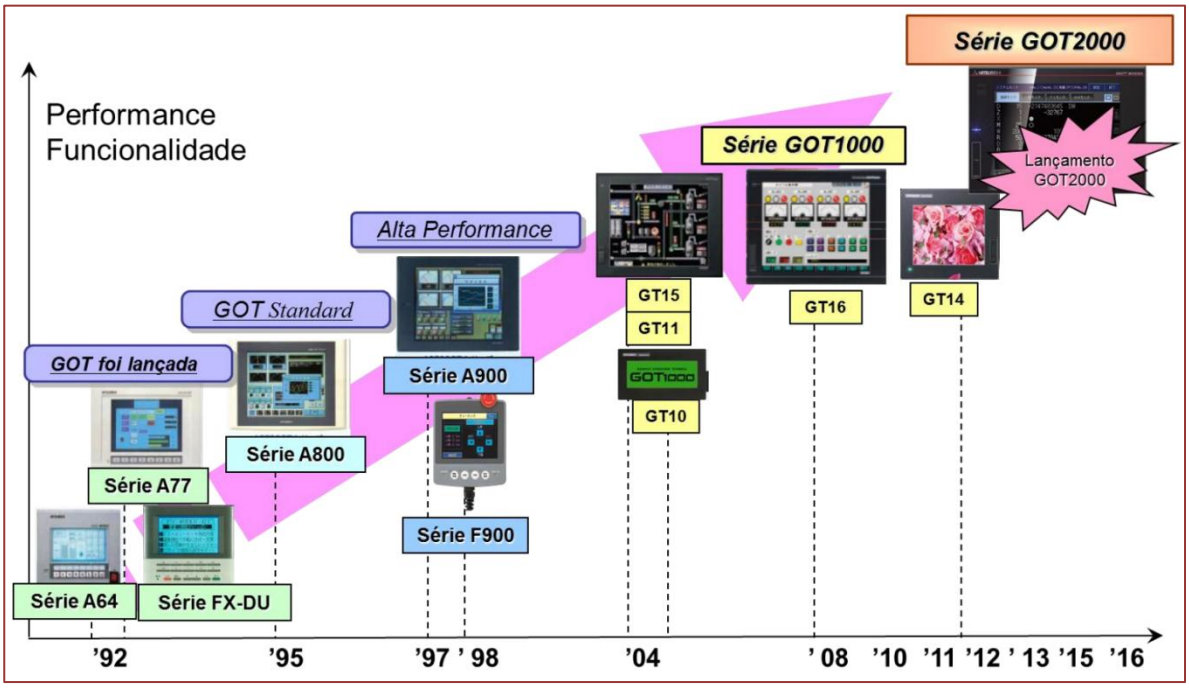
Uma pergunta frequente que é feita pelos alunos é sobre a possibilidade da troca de uma IHM por um monitor de computador ou algo do tipo, a resposta é que se o ambiente não for hostil dentro de determinadas situações pode ser possível, contudo o ambiente fabril é sempre considerado hostil, assim as IHMs são projetadas para trabalharem várias horas seguidas, com diversos níveis de poeira, humidades, ruídos variações de tensão *etc*, esses são mais alguns diferenciais para usamos essas IHMs projetadas para esse ambiente fabril, é mais ou menos a diferença de um computador pessoal comum e um computador industrial.

7.2.2. HISTÓRICO DAS IHM

Definir quando começou e quem foi o primeiro fabricante das ditas IHMs modernas, é complicado, pois deveríamos definir exatamente o que é uma IHM moderna, pois as primeiras IHM não tinham *touch* nas suas telas, os meios de comunicações eram mais simples, ou seja, uma série de tecnologias foram e vem sendo incorporadas ano a ano. Dessa forma, como estamos trabalhando com as IHMs da MITSUBISHI (Os GOTs), vamos falar brevemente do histórico desse fabricante.

Uma das primeiras IHMs da MITSUBISHI foi às da série A64 pelos anos de 1990. Era bem rudimentar sobre vários aspectos tecnológicos atuais. Na Figura 7.2 é possível observar um gráfico dos modelos da MITSUBISHI ao longo dos anos.

Figura 7.2 – Modelos de IHMs da Empresa Mitsubishi



Fonte: Mitsubishi 01, 2020.

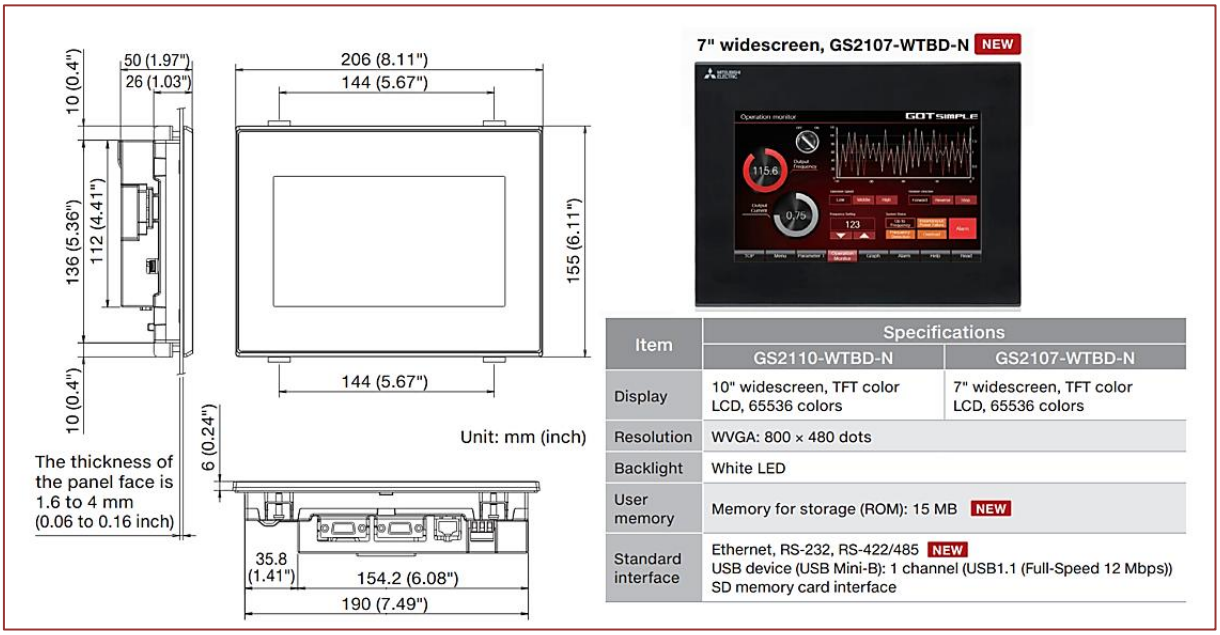
Portanto, o GOT foi lançado nos anos 90, e podemos definir essas IHMs em 3 grupos básicos: a GOT *Standard*, a GOT1000 e por último e mais atual a GOT2000.

O modelo da GOT que usamos para as práticas e projetos foi da GOT2000, contudo uma linha de baixo custo fabricada na China, que embora seja do grupo mais moderno (GOT2000) vem com menos recursos, por isso é chamada de GOT *SIMPLE Series*.

7.2.3. DESCRIÇÃO DO *HARDWARE* IHM, GOT *SIMPLE SERIES* (GS21)

Essa serie simples da GOT2000, a GS21, pode ser comprada na tela de 10 ou 7 polegadas, possui tela com *touch* (algumas IHMs da serie GOT2000 tem *multitouch*), resolução de 800x480 pixels, com tecnologia LCD TFT de 65536 cores, *backlight* de LED branco, Memória ROM interna até 15 MB, alimentação de energia de 24 V DC, entrada de cartão de memória tipo SD, interfaces de comunicação com outros dispositivos: Ethernet, RS-232 e RS-422/485, e USB. Na Figura 7.3 é possível observar algumas dessas algumas dessas características. Nessa figura também é possível verificar as dimensões mecânicas da tela de 7 polegadas, que foi a que usamos nessa disciplina.

Figura 7.3 – Detalhes da IHM GOT2107-WTBD-N

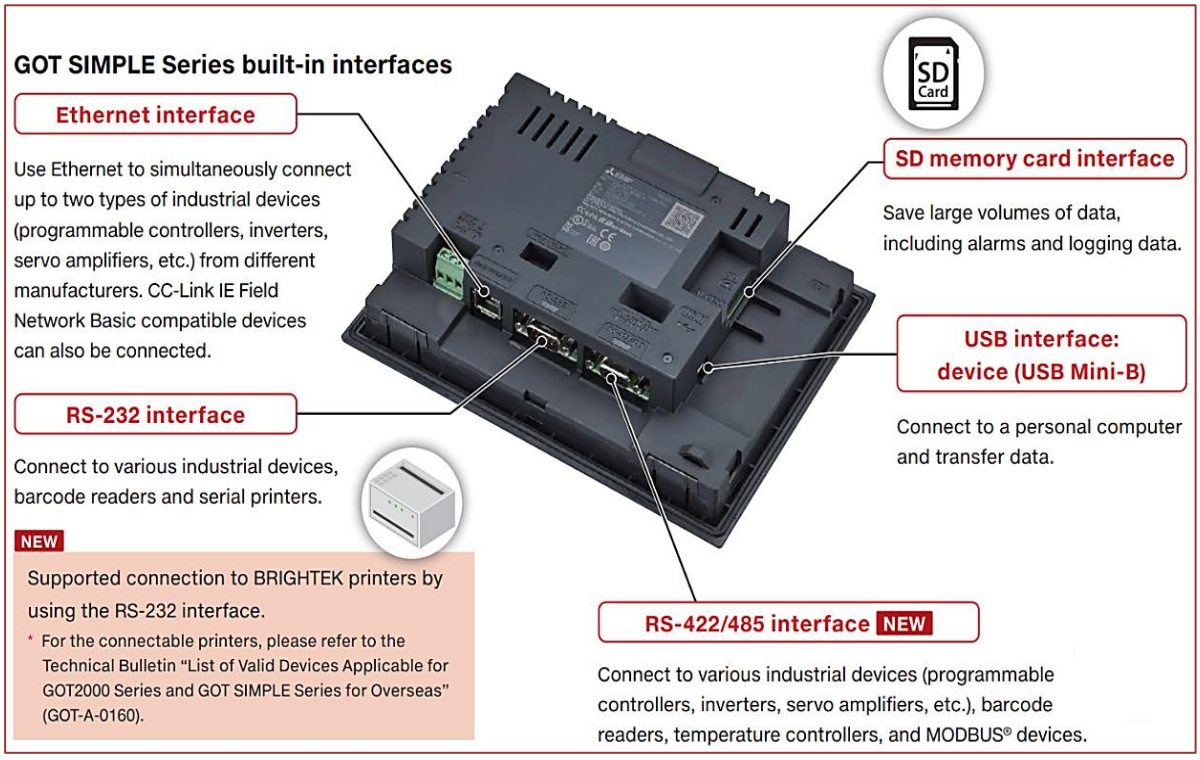


Item	Specifications	
	GS2110-WTBD-N	GS2107-WTBD-N
Display	10" widescreen, TFT color LCD, 65536 colors	7" widescreen, TFT color LCD, 65536 colors
Resolution	WVGA: 800 × 480 dots	
Backlight	White LED	
User memory	Memory for storage (ROM): 15 MB NEW	
Standard interface	Ethernet, RS-232, RS-422/485 NEW USB device (USB Mini-B): 1 channel (USB1.1 (Full-Speed 12 Mbps)) SD memory card interface	

Fonte: Editado de Mitsubishi 03, 2021.

Na Figura 7.4 tem-se a imagem da parte traseira da IHM GOT2107-WTBD-N, onde são localizadas suas interfaces.

Figura 7.4 – Detalhes da IHM GOT2107-WTBD-N, parte traseira



Fonte: Mitsubishi 03, 2021.

7.2.4. TIPOS DE CONEXÃO – IHM (GOT) E CLP

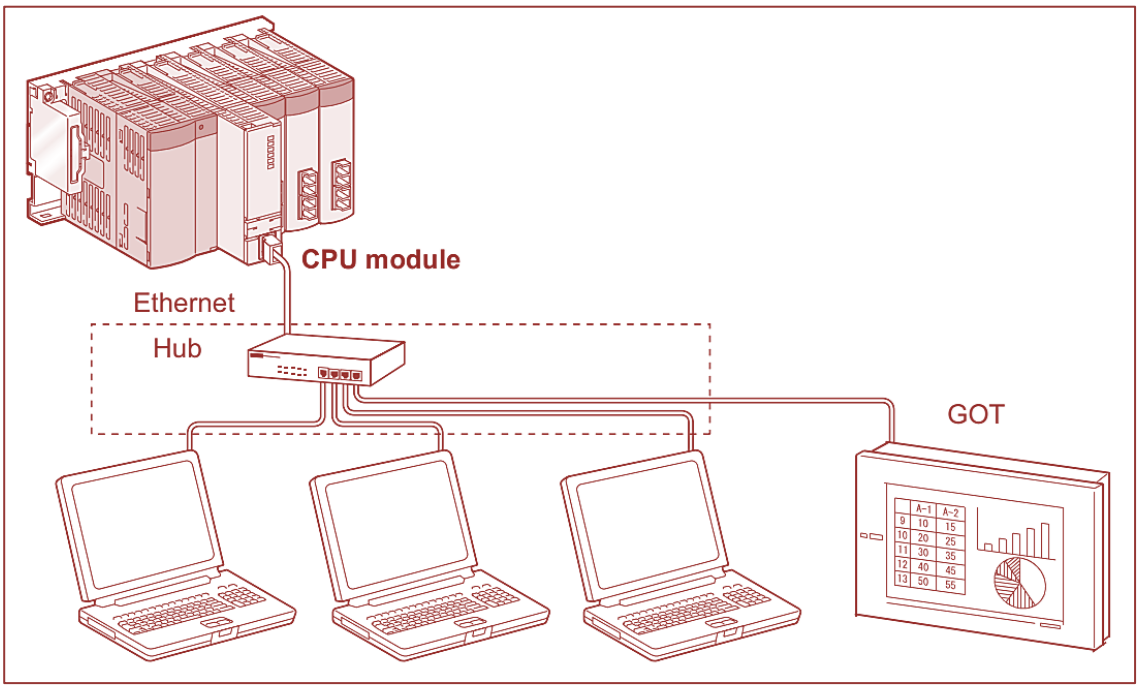
Existe uma grande variedade de tipos diferentes de conexões (topologias) entre uma IHM e um CLP, uma grande variedade dessas topologias é apresentada aos alunos, contudo aqui vamos destacar apenas a que foi mais usada nas práticas dessa disciplina.

Na Figura 6.5 é mostrada essa conexão na topologia estrela. Essas conexões podem usar diversos tipos de cabos e conectores, bem como vários tipos de protocolos de comunicação industriais, contudo foi usado preferencialmente o padrão Ethernet e uma variação do protocolo CC-Link que usa o padrão Ethernet.

O desenvolvimento do projeto é feito em duas etapas, a primeira na bancada do laboratório LRI e a segunda etapa na célula robótica.

Primeira etapa: Inicialmente é ligado ao HUB (ou Switch Não Gerenciável) que fica nas bancadas do laboratório uma IHM e um CLP, depois os computadores para o desenvolvimento do projeto, programação da IHM e do CLP, a quantidade de computadores depende da quantidade de alunos por bancada. Conforme pode ser visto na figura 3.5. Assim todos os alunos daquela bancada podem compartilhar o uso da IHM e do CLP, podendo fazer a edição, compilação, simulação, programação, monitoramento e ensaios dos projetos desenvolvidos. É nessa etapa que feita boa parte das explicações do professor usando vários recursos didáticos como projetor de imagem (*115atashow*), lousa branca dentre outros. Vale destacar que os alunos da mesma bancada fazem parte do mesmo projeto final a ser desenvolvido, fomentando assim a interação entre os mesmos.

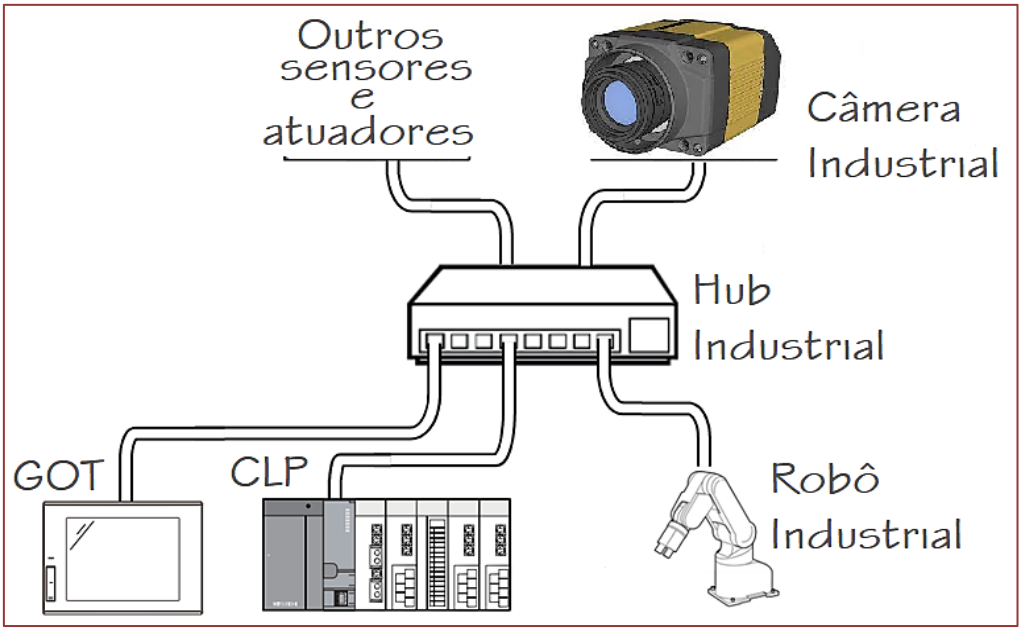
Figura 7.5 – Primeira Etapa: Desenvolvimento nas bancadas



Fonte: Editado de Mitsubishi 04, 2018.

Segunda etapa: Essa já na célula robótica os alunos vão fazer a integração do CLP e da IHM (GOT), com outros dispositivos como: o robô industrial, câmera industrial, e outros sensores e atuadores, conforme pode ser visto na Figura 7.6.

Figura 7.6 – Segunda Etapa: Desenvolvimento nas Células Robóticas



Fonte: Editado de Mitsubishi 04, 2018.

É claro que muitos sensores e atuadores podem ser ligados diretamente nos cartões (módulos) entradas e saídas do CLP e comandados pela IHM, inclusive os próprios robôs por meio das suas controladoras podem fazer essa comunicação direta mais simples. Como exemplo de sensores ligados ao conjunto CLP/IHM podemos destacar: câmera industrial da COGNEX, sensor de presença capacitivo e indutivo (tanto analógico e digital), sensor de pressão *etc.* Os principais atuadores na célula robótica além do robô são as ferramentas ligadas a ele como ventosas e garras, essas ferramentas são acionadas eletricamente ou por meio de válvulas eletropneumáticas, e o motor da esteira controlado diretamente por um inversor de frequência ligado ao conjunto CLP/IHM.

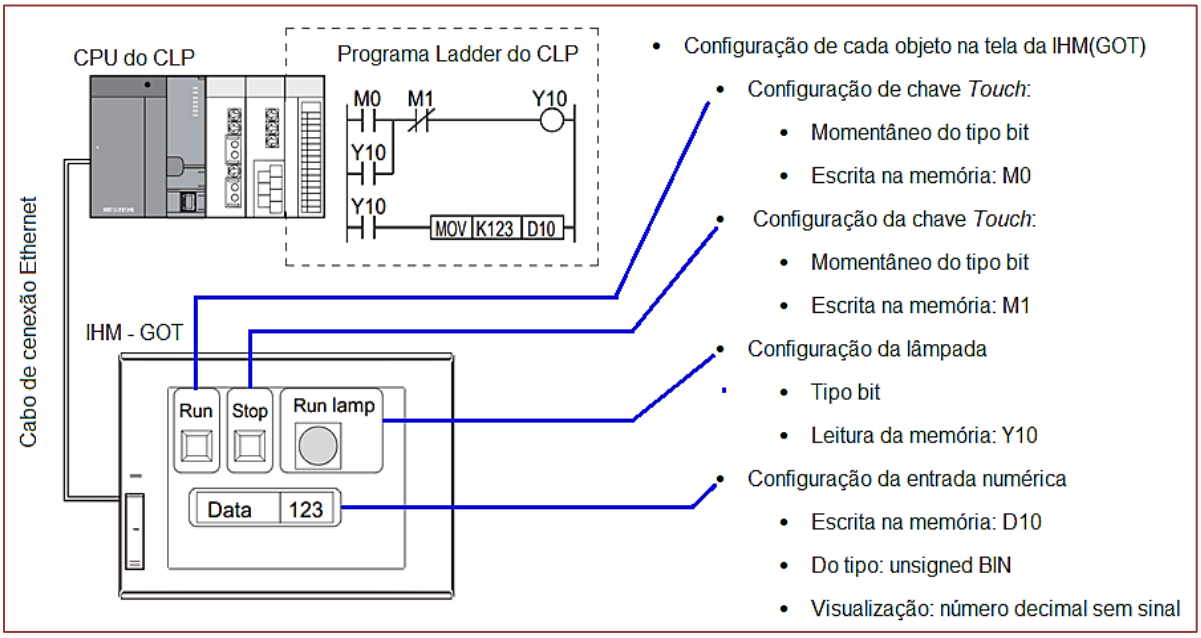
7.2.5. PROGRAMAÇÃO E DESENVOLVIMENTO DA IHM (GOT)

Com o *software* de desenvolvimento GT Designer3, é feita a construção de telas na IHM da empresa MITSUBISHI, bem como as figuras do painel de exibição chamadas objetos, como interruptor de toque, lâmpada e display numérico, *etc*. A IHM (GOT) permite a leitura e gravação em dispositivos de memória internos do CLP (dos tipos *bit* e *word*) e são associados com os vários objetos criados nas telas da IHM(GOT) como interruptores, lâmpadas, entrada/saídas numérica e em texto, exibição de mensagens, etc.

A seguir é feita uma pequena demonstração de um projeto do CLP/IHM. São criados os objetos em uma tela da IHM, chamados Run, Stop, Run lamp e Data. Esses objetos são associados a elementos de memória do CLP e configurados conforme pode ser visualizado na Figura 7.7. Observe que alguns desses objetos são de escrita outros de

leitura dessas memórias internas do CLP (cada tipo desses de memória do CLP a MITSUBISHI chama de *device*).

Figura 7.7 – Demonstração de Funcionamento de um Projeto CLP/IHM



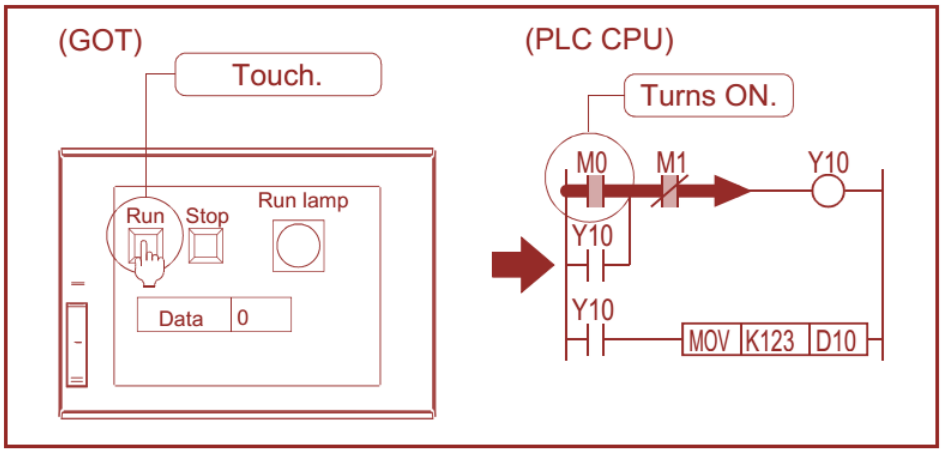
Fonte: Editado de Mitsubishi 04, 2018.

É possível visualizar também na figura 6.7 um pequeno código em Ladder sendo executado na CPU do CLP.

Descrição das ações:

1) Acionamento da chave Run: Enquanto o botão (chave) de toque "Run" no GOT estiver sendo tocado, o contato do tipo bit "M0" no programa Ladder do CLP estará fechado, conforme é possível visualizar na Figura 7.8.

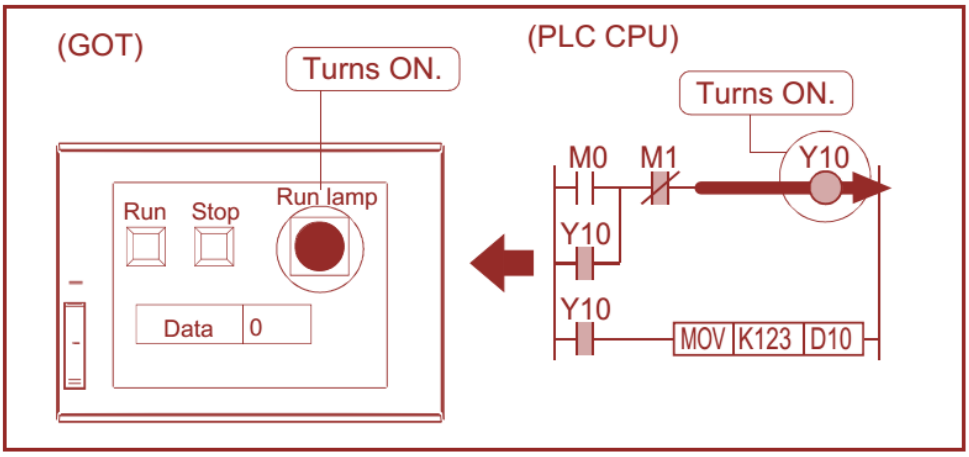
Figura 7.8 – Acionamento da chave Run



Fonte: Mitsubishi 04, 2018.

2) Saída Y10 ligada: A consequência imediata após o contato "M0" ser fechado, é o acionamento do dispositivo de saída "Y10", conforme é possível visualizar na Figura 7.9. Além disso, a figura Run Lamp na IHM(GOT) é modificada, ilustrando uma lâmpada ligada (em ON), pois ela foi associada anteriormente ao dispositivo de saída "Y10". Ou seja, se a saída Y10 do CLP é ativada a figura Run Lamp na IHM é modificada, ilustrando uma lâmpada ligada.

Figura 7.9 – Saída Y10 do CLP é ligada, Lâmpada da IHM é acesa e o Selo é Fechado

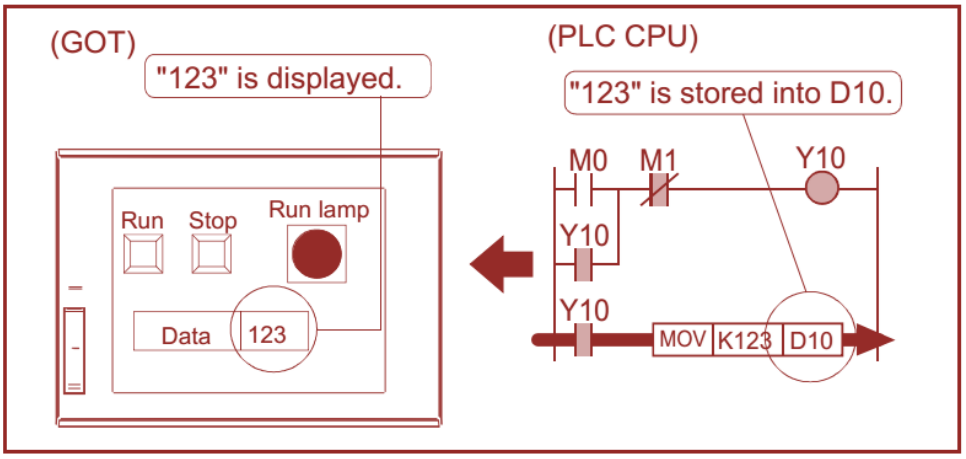


Fonte: Mitsubishi 04, 2018.

3) Selo Acionado: Quando a chave Run (na IHM(GOT)) não é mais acionada a saída Y10 continua sendo ativo pelo contato de selo Y10, a saída Y10 só pode ser desligada agora abrindo o contato M1(que é normalmente fechado). Não confundir a saída Y10 com o contado Y10, embora os dois estejam associados são coisas distintas. Conforme é possível visualizar também na Figura 7.9.

4) Ativação da instrução MOV: Outra consequência imediata do acionamento da saída Y10 é a ativação da instrução MOV por meio do contado Y10 na última linha do programa em Ladder. Essa instrução movimenta (Escreve, Grava) no dispositivo de memória interna do CLP D10 (que é do tipo word) o valor em decimal cento e vinte e três (123). É claro que 123 também é exibido no display numérico Data da IHM(GOT) no qual o dispositivo monitor foi definido para o dispositivo de palavra "D10". Conforme é possível visualizar também na Figura 7.10

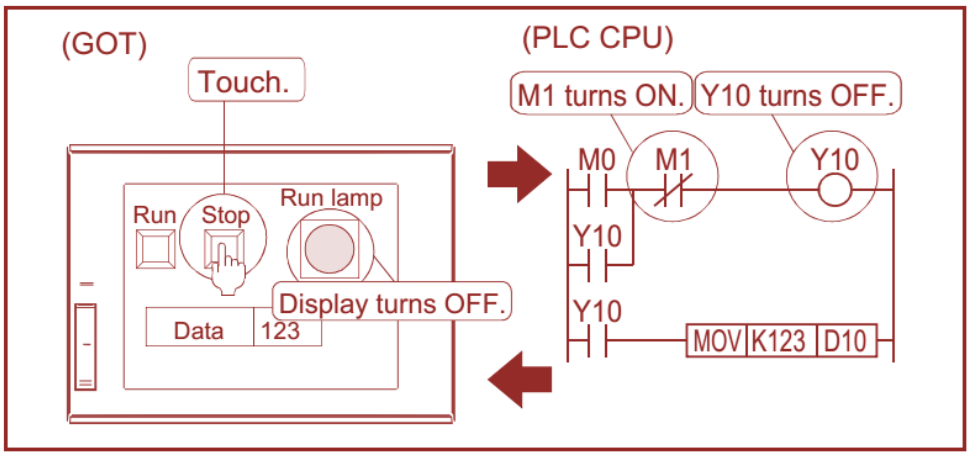
Figura 7.10 – Ativação da Instrução MOV



Fonte: MITSUBISHI 04, 2018.

5) Acionamento do Stop: Enquanto a chave de toque "Stop" no GOT não for tocada, o dispositivo de bit do PLC "M1" fica ligado. Quando é tocado na IHM o botão Stop o dispositivo do PLC "Y10" é DESLIGADO e a Lâmpada de funcionamento (Run Lamp) é DESLIGADA. Conforme é possível visualizar também na Figura 7.11.

Figura 7.11 – Acionamento do Stop



Fonte: Mitsubishi 04, 2018.

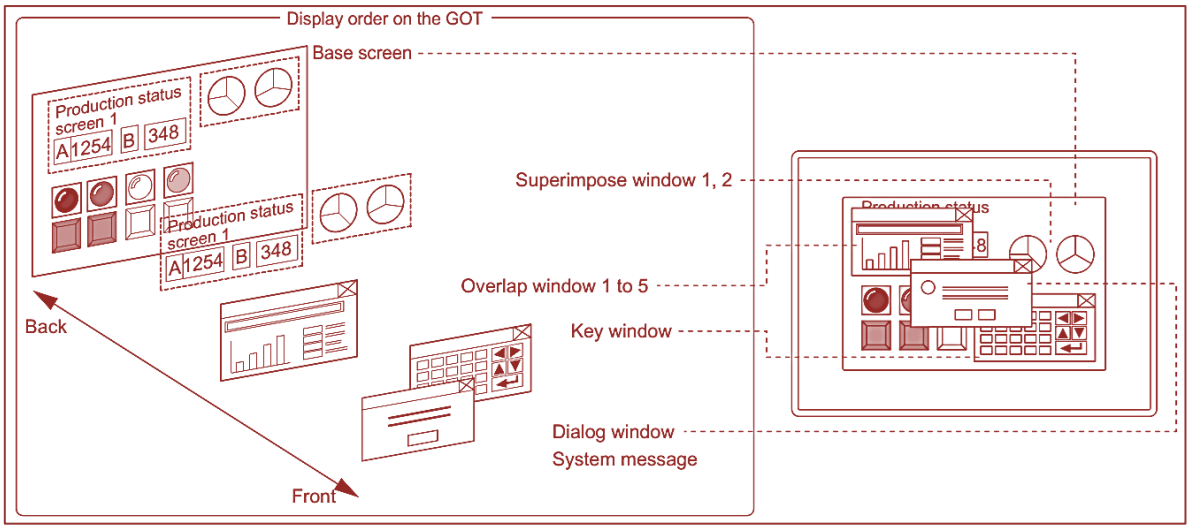
Antes de falarmos sobre alguns objetos como *Switch*(chaves) *e Lamp*(lâmpadas), vamos explicar sobre as Telas (Screens) da IHM.

As telas da IHM podem ser do tipo Base Screen e Window Screen. As Base Screen são caracterizadas por obrigatoriamente ocupar toda a área do Display da IHM(GOT), e é possível e bem comum criar várias dessas telas, com botões (Switch) criados e dedicados para navegação entre essas telas. As Window Screen são caracterizadas por não obrigatoriamente ocupar toda a área do Display da IHM, podem ser de 3 tipos, a saber:

- Overlap Window: São pop-up's que possui um X no canto superior direito, usado para fechar a janela. Podendo criar objetos como botões e lâmpadas como em feito em qualquer tela.
- Superimpose Window: São pop-up's (semelhante ao Overlap Window) sem o X, devendo ser criados pelo programar mecanismos para ser possível fechar essa janela, como não possui o X, esses mecanismos podem ser feito no CLP ou mesmo na IHM. Assim geralmente é criado alguns botões nessa janela impondo o operador a escolher uma das opções.
- Dialog Window: São pop-up's (semelhante ao Overlap Window) que serve apenas para dar um aviso (informação).

Essas telas e Janelas tem uma ordem de apresentação no Display da IHM conforme pode ser observado na Figura 7.12

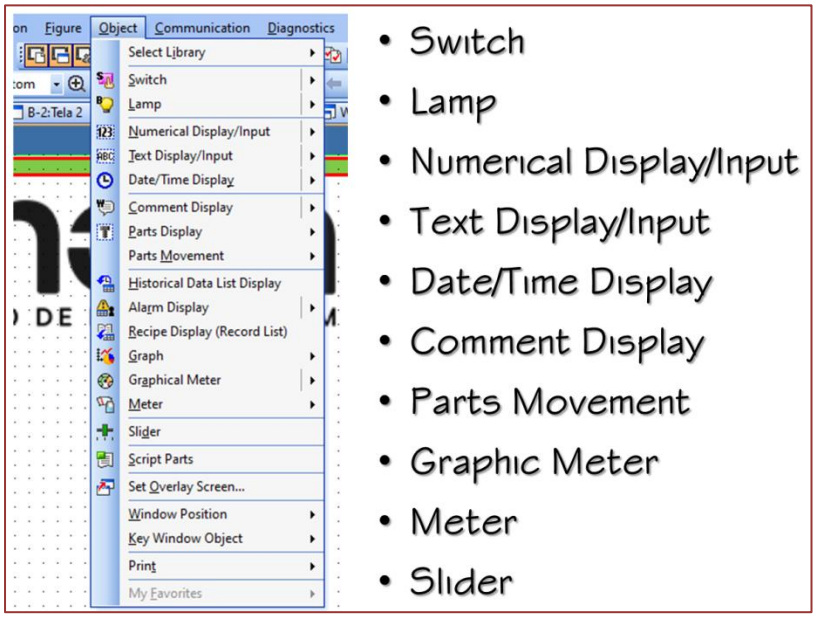
Figura 7.12 – Ordem de Apresentação das Telas e Janelas na IHM GOT2000



Fonte: Mitsubishi 02, 2015.

Em seguida é mostrado na Figura 7.13, uma lista dos nomes dos principais objetos para criação das telas da IHM (GOT) e onde eles podem ser localizados no *Software* GT Designer3 da Empresa Mitsubishi.

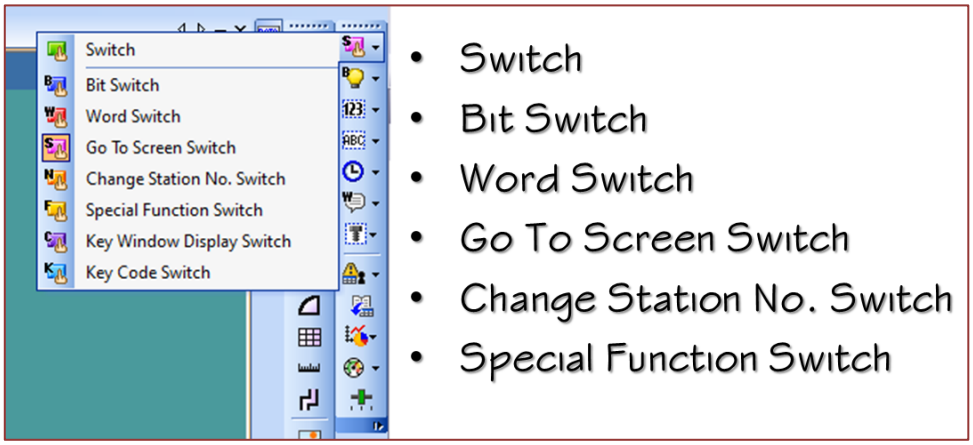
Figura 7.13 – Principais objetos usados na criação das telas da IHM



Fonte: Os autores, 2023.

Cada um desses objetos possui uma série de variações, como exemplo é mostrado na Figura 7.14 as variações do objeto Switch(chave).

Figura 7.14 – Variações, ou tipos distintos do objeto Switch



Fonte: Os autores, 2025.

É claro que existe uma grande variedade de outras configurações (funções) fora os objetos, Telas e Janelas que podem ser utilizadas em uma programação de uma IHM, tais como: Alarmes, Senhas, Receitas, Arquivos de *logs*, Animações *etc*.

7.2.6. DESENVOLVIMENTO DO CASE DIDÁTICO PARA IHM

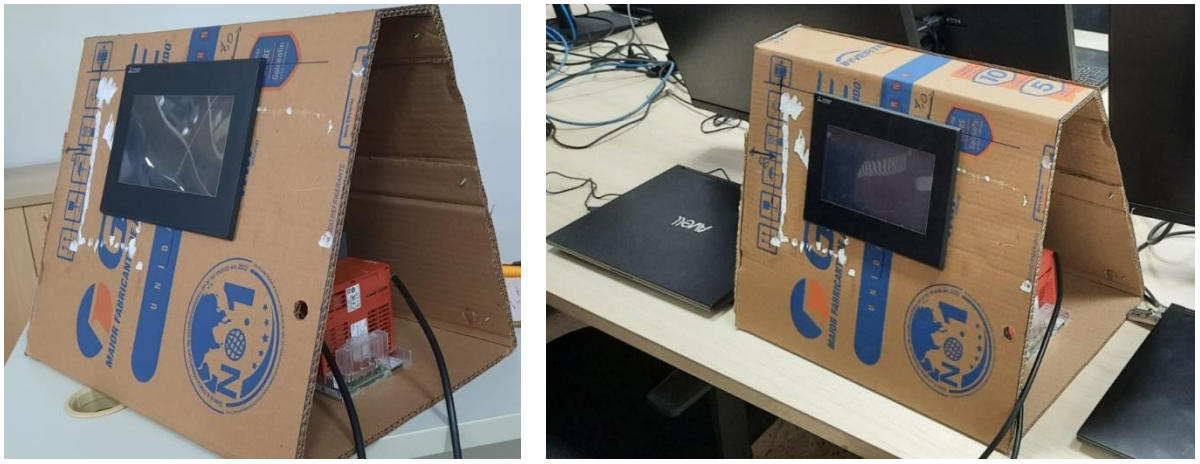
O processo de ensino-aprendizagem em automação industrial pode ser amplamente enriquecido quando o estudante participa não apenas da programação e operação de equipamentos, mas também do processo de concepção e construção das interfaces físicas. Nesse contexto, o projeto do *case* didático para a Interface Homem-Máquina (IHM) do curso de capacitação avançada em Robótica Industrial foi desenvolvido com o objetivo de oferecer uma estrutura modular, portátil e de baixo custo, que permitisse a integração de diversos componentes da automação, como CLPs, fontes, switches e IHMs.

7.2.6.1. IDEAÇÃO E PROTOTIPAGEM INICIAL

O projeto teve início com uma etapa de ideação, na qual foram discutidos aspectos funcionais e ergonômicos do suporte da IHM. Buscou-se um formato que proporcionasse facilidade de montagem, visualização e transporte, considerando o uso em ambiente educacional.

Como parte do processo de prototipagem rápida, a primeira versão do *case* foi construída em papelão reciclado, simulando o formato e a inclinação ideais para o painel da IHM. Essa abordagem de baixo custo possibilitou ajustes rápidos de dimensões, recortes e posicionamento dos componentes eletrônicos internos, como demonstrado na Figura 7.15.

Figura 7.15 – *Case* Didático para IHM – Ideação em Papelão



Fonte: Os autores, 2025.

7.2.6.2. TRANSIÇÃO PARA O MODELO FINAL

Após os testes de posicionamento e validação ergonômica, iniciou-se a fase de modelagem digital do case em acrílico, utilizando softwares de CAD para o detalhamento das dimensões e dos recortes necessários para os dispositivos. As peças foram cortadas em máquina CNC a laser, garantindo precisão e acabamento profissional, conforme pode ser visto na Figura 7.16.

Figura 7.16 – *Case* Didático para IHM – Corte das peças de acrílico em CNC a laser

Fonte: Os autores, 2025.

Além do acrílico, foram projetadas e impressas em 3D as peças de fixação e união entre as chapas (peças de acrílico), conforme pode ser visto na Figura 7.17. Proporcionando uma estrutura robusta e modular. Essa combinação de manufatura aditiva e subtrativa refletiu diretamente os conceitos de Indústria 4.0, aplicados em um contexto educacional.

Figura 7.17 – *Case* Didático para IHM – Impressão 3D das peças de fixação e união

Fonte: Os autores, 2025.

7.2.6.3. APLICAÇÕES EDUCACIONAIS

O *case* final, conforme pode ser visto na Figura 7.18, permite que os estudantes conectem e testem, de forma segura e organizada, diferentes topologias de rede industrial e configurações de comunicação entre a IHM e o CLP. Essa estrutura favorece o aprendizado prático de integração de sistemas, estimulando a experimentação e a autonomia dos alunos.

Além disso, o processo completo — desde a prototipagem em papelão até a montagem final em acrílico — serviu como um exercício interdisciplinar, conectando conceitos de design, manufatura digital e automação industrial.

Figura 7.17 – *Case* Didático para IHM – Montagem final

Fonte: Os autores, 2025.

7.3. PROCESSO AVALIATIVO

As avaliações foram realizadas observando o cumprimento das atividades (pequenos projetos de fixação) durante as aulas e ao final com apresentação do projeto da IHM de cada equipe, muitas dessas atividades são feitas em equipe fomentando a integração dos membros de cada grupo.

Um dos propósitos dessas atividades avaliativas é a construção em etapas do seu projeto final do curso. Assim, o critério para correção é baseado principalmente em aspectos qualitativos e de forma secundária os quantitativos, valorizando a criatividade e pontualidade.

REFERÊNCIAS

[1] MITSUBISHI 01, A. Apostila: **Treinamento Básico GOT2000**. [*S. l.*]: MITSUBISHI, 2020.

[2] MITSUBISHI 02. **GT Designer3 (GOT2000) Screen Design Manual**. [*S. l.: s. n.*], 2015.

[3] MITSUBISHI 03. Manual MITSUBISHI ELECTRIC CORPORATION: **Graphic Operation Terminal GOT SIMPLE Series**. [*S. l.: s. n.*], 2021.

[4] MITSUBISHI 04. **Graphic Operation Terminal (GOT2000 Series) Basic Training Textbook** (for GT Works3). [*S. l.*]: MITSUBISHI ELECTRIC CORPORATION, 2018.

CAPÍTULO

08 Visão computacional e aplicações industriais

Marcelo Chamy Machado[3], Anderson Gadelha Fontoura[4]

8.1. CARACTERIZAÇÃO DA DISCIPLINA

A disciplina de Visão Computacional foi elaborada de forma a possibilitar que todos os alunos, mesmo os que não tinham conhecimento prévio dos conceitos, técnicas, ou embasamento necessário na linguagem de programação utilizada, pudessem compreender, aplicar e codificar cada assunto. O método hands-on foi utilizado de forma que, a cada tópico do conteúdo apresentado, os conceitos mais importantes eram detalhados e em seguida era feita a codificação, na qual os programas eram criados passo a passo na linguagem Python, com a explicação de cada comando e funcionalidade abordada com um ou mais exemplos práticos, e cada aluno replicava 100% dos códigos. Após essa etapa, algumas atividades práticas eram indicadas, e os alunos passavam a elaborar os programas e realizavam discussões entre as equipes, para entender onde cada técnica poderia ser aplicada nas possíveis soluções para os projetos. Devido à quantidade de assuntos necessários, a elaboração do conteúdo da disciplina visou a aplicação prática de alguns dos principais conceitos, sem aprofundamento dos fundamentos matemáticos e de álgebra.

8.1.1. EMENTA

Conceitos de ambientes virtuais e configuração de ambientes para programação em Python; Introdução à linguagem Python; Introdução à visão computacional; Fundamentos de processamento de imagens e vídeo e técnicas tradicionais de visão computacional; Aprendizado de máquina e aplicações em visão computacional; Aplicação de bibliotecas e ferramentas para visão computacional; Treinamentos de modelos de forma remota e avaliação de resultados dos modelos; Aplicações industriais; Elaboração de projetos práticos no contexto industrial;

8.1.2. OBJETIVOS

Capacitar os alunos para compreender os principais conceitos e aplicar métodos, técnicas e estratégias correlatas à visão computacional e processamento de imagens em ambientes industriais.

[3] Mestre em Ciência da Computação. Professor da disciplina de Visão Computacional. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). E-mail: *marcelo.chamy@ifam.edu.br.*
[4] Mestre em Ciência da Computação. Professor da disciplina de Visão Computacional. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). E-mail: *anderson.fontoura@ifam.edu.br*

Objetivos específicos:

- Desenvolver algoritmos básicos, intermediários e avançados na linguagem de programação Python;
- Apresentar e implementar os principais algoritmos de processamento digital de imagem, de visão computacional e de aprendizagem de máquina utilizados com visão computacional;
- Treinar modelos de aprendizado de máquina e avaliar os resultados de forma crítica e consistente;
- Avaliar os desafios associados à operação em tempo real, considerando fatores de latência, processamento e envio de sinal aos robôs.

8.1.3. CARGA HORÁRIA

Carga Horária Teórica: 30 h

Carga Horária Prática: 50 h

8.2. CONTEÚDOS

Essa seção descreve os conteúdos aplicados no decorrer da disciplina.

8.2.1. INTRODUÇÃO À LINGUAGEM PYTHON

A linguagem Python foi criada por Guido van Rossum em 1989 e lançado publicamente em 1991, com o objetivo de ser uma linguagem simples, legível e poderosa. Python é uma das linguagens fundamentais para lidar com problemas de Visão Computacional, devido à sua simplicidade, baixa curva de aprendizado e versatilidade, com uma ampla gama de bibliotecas com muitos métodos já implementados, possibilitando com isso a redução do tempo necessário para desenvolvimento de soluções robustas e confiáveis.

Python é uma linguagem de programação de alto nível, interpretada e de tipagem dinâmica, criada para ser simples e legível, permitindo que desenvolvedores escrevam código de forma clara e concisa. Ela suporta múltiplos paradigmas, como programação orientada a objetos, funcional e imperativa, e possui uma vasta biblioteca padrão que facilita tarefas comuns sem necessidade de reinventar soluções.

Python é usado de forma bastante simples, bastando instalar a linguagem no computador, escolher um editor ou ambiente de desenvolvimento (como VS Code, PyCharm ou Jupyter Notebook), escrever os programas em arquivos com extensão .py e os executar pelo terminal ou diretamente no editor. A linguagem permite criar desde scripts básicos, como imprimir mensagens na tela, até aplicações complexas, aproveitando sua vasta biblioteca padrão e pacotes externos que podem ser instalados com o gerenciador pip. Em resumo, usar Python significa escrever código claro e legível, rodar esse código com o interpretador e expandir suas possibilidades com bibliotecas para diferentes áreas como web, ciência de dados, automação e inteligência artificial.

Para realização das atividades do curso, foram configurados ambientes virtuais com a utilização do pacote “Anaconda Python”, conforme Vaughan (2023), em todas as

máquinas, com os alunos realizando sob orientação a criação de um novo ambiente, e após a sua ativação, foram instalados todos os pacotes que seriam necessários no decorrer da disciplina, sendo eles basicamente: OpenCV-Python, Tensorflow, Keras, Matplotlib, Numpy, Scikit-Learn e Jupyter Notebook. Após isso, as atividades práticas foram iniciadas com a utilização da IDE Jupyter Notebook, escolhida devido à sua facilidade e possibilidade de manter um histórico das execuções, além de manter todas as saídas dos comandos ao salvar o arquivo, facilitando revisões e correções de atividades.

Foram feitas revisões de tópicos essenciais de programação em Python e realizadas atividades no formato hands-on, juntamente com alguns exercícios para que os alunos pudessem se ambientar com a linguagem: Indentação; variáveis e tipos de dados; operações de entrada e saída; controle de fluxo, estruturas de repetição com While e For; Criação de Funções; Operações com Coleções; Numpy; Matplotlib;

Após essa etapa inicial, foi dado início aos tópicos referentes a processamento digital de imagens e visão computacional, conforme descrito a partir da Seção 8.2.2.

8.2.2. INTRODUÇÃO À VISÃO COMPUTACIONAL

Visão Computacional é um campo interdisciplinar e refere-se basicamente à criação de programas capazes de interpretar, ou dar sentido ao que imagens representam. A diferença entre visão computacional e processamento de imagens é que as técnicas de processamento de imagens podem ser utilizadas para realizar adequações e melhorias, e a partir disso, extrair características que são importantes para dar significado ao que existe em imagens.

Para que seja possível aplicação da visão computacional, é necessário combinar conhecimentos de várias áreas tais como:

- Processamento de imagem e vídeo: aplicação de técnicas fundamentais e essenciais para que um sinal seja processado e melhorado, possibilitando a extração de informações relevantes para o problema em questão;
- Física: entendimento de conceitos básicos de aquisição de imagens utilizando câmeras, tipos de lente, tipos de iluminação e características importantes e influência da luz na geração de imagens;
- Ciência da Computação: conhecimento e utilização de algoritmos e estruturas de dados adequadas para possibilitar que a aplicação das técnicas seja realizada com um bom desempenho;
- Aprendizado de Máquina: aplicação de técnicas envolvendo modelos que consigam aprender, classificar e auxiliar na tomada de decisão;
- Robótica: utilização do resultado final da interpretação da imagem (tomada de decisão) no envio de comandos para robôs executarem tarefas para os quais foram previamente programados.

Existem algumas divisões na Visão Computacional, relacionadas diretamente com a aplicação para qual uma solução está sendo elaborada. Alguns exemplos que podem ser obtidos a partir da combinação de conhecimentos citados anteriormente são:

- Medições: comparações de tamanho de objetos tendo um modelo prévio como referência, ou confirmações de medidas de objetos, utilizando proporções pré-

determinadas e uma tolerância que também pode ser especificada previamente. Exemplos de medição são: identificar distância entre determinados componentes presentes em determinado circuito, ou o tamanho / espessura de algum componente ou produto produzido;

- Contagem: identificação do número (quantidade) de partes, ou a quantidade de determinada característica de determinadas partes, com por exemplo: confirmar se a quantidade de furos identificados em determinado componente está conforme o esperado; contar o número de determinados itens em uma peça, e dessa forma, pode-se identificar quantos itens estão ausentes;
- Localização: indicação de posição ou orientação de uma peça ou componente. Como exemplo, pode ser citado: detecção de componentes desalinhados ou mal encaixados; possibilidade de reconhecimento de determinadas características, mesmo que os objetos estejam desalinhados, como por exemplo, realizar o reconhecimento de um padrão;
- Leitura de padrões: leitura de códigos numéricos ou alfanuméricos através de reconhecimento ótico de caracteres (OCR) e leitura de dados através de códigos de barras e *QR-Codes*. A partir da identificação do código, pode-se definir os tipos de padrão esperados do produto em produção.
- Detecção de anomalias: a partir de um ou mais modelos pré-determinados, pode-se detectar se o objeto ou produto contém alguma não conformidade, como por exemplo, arranhões, cores diferentes do padrão esperado, partes quebradas, e inúmeras outras situações.

É normal que determinadas aplicações de visão computacional necessitem da combinação de vários desses itens relacionados, dependendo da complexidade do problema e quantidades de etapas necessárias para automatizar processos de identificação em imagens ou vídeo.

Esses processos são muito presentes em indústrias de vários tipos, e o objetivo da aplicação de técnicas automatizadas é basicamente a otimização de etapas ineficientes, redução de defeitos e rastreamento, muitas vezes realizadas por seres humanos através de inspeção visual. Por mais eficientes e eficazes que sejam em atividades repetitivas e padronizadas, não podem ser comparados com as máquinas e algoritmos. Estes possuem diversas vantagens, tais como: trabalho de forma ininterrupta sem paralização por cansaço, férias, doenças; maior velocidade, que pode ser adaptada com a aplicação de equipamentos tecnologicamente cada vez mais avançados; e consistência, sem subjetividade para a tomada de decisão.

8.2.3. FUNDAMENTOS DE PROCESSAMENTO DE IMAGENS E VÍDEO E TÉCNICAS TRADICIONAIS DE VISÃO COMPUTACIONAL

O passo inicialmente necessário para abordar tópicos de visão computacional é entender o processo de aquisição de uma imagem, quais características são importantes considerar, e por fim, quais são as técnicas que podem ser aplicadas e suas finalidades.

Uma imagem é uma visão bidimensional (2D) gerada a partir de uma cena real em um mundo tridimensional. É formada a partir da captura (aquisição), processada por uma câmera. Uma vez que é feita a aquisição da imagem de um objeto ou cena, é gerado um conjunto de pixels com intensidades variadas. A quantidade de pixels é relacionada com

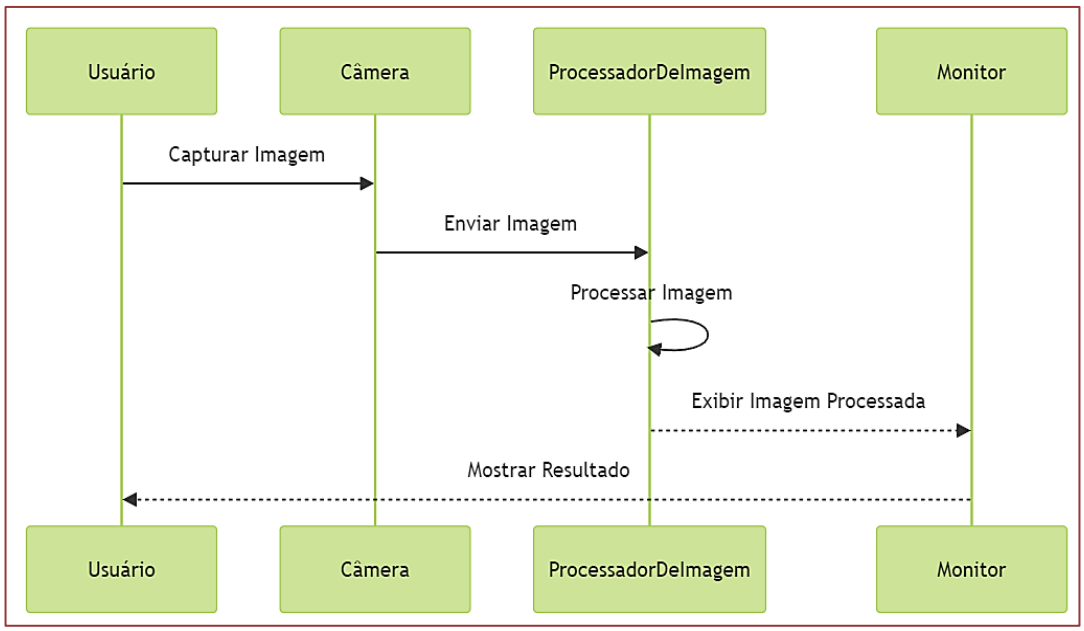
a resolução da captura (quantidade de linhas x quantidade de colunas), e a intensidade de cada pixel é resultante da intensidade de luz refletida nos objetos capturados.

Objetos existentes têm como resultados imagens diferentes, dependendo da perspectiva de captura. Além disso, existem fatores não relacionados com a construção do dispositivo de captura que afetam o processo de formação de imagens e vídeos, e a qualidade dos arquivos resultantes. Estes fatores são chamados de **extrínsecos**, e incluem:

- Iluminação, tais como tipo, intensidade, direção, cor e uniformidade da luz;
- Posição e orientação da câmera: distância da câmera para o objeto, assim como o ângulo da lente para o alvo afetam a perspectiva, o enquadramento e a composição da imagem;
- Movimentação: qualquer movimentação da câmera ou do objeto pode causar perda de foco e distorções;
- Condições ambientais: temperatura, umidade e poluição afetam a qualidade da imagem; e
- Interferências externas: refrações, reflexos ou obstruções ou sobreposições alteram a aparência dos objetos.

O processamento de imagens consiste normalmente das seguintes etapas: a captura da imagem pelo usuário, o processamento da imagem (que pode ser repetido ou combinado com diversas técnicas) e geração do resultado, para ser mostrado para o usuário. A Figura 8.1 apresenta um diagrama que ilustra esse processo.

Figura 8.1 - Etapas para aplicação do processamento de imagens

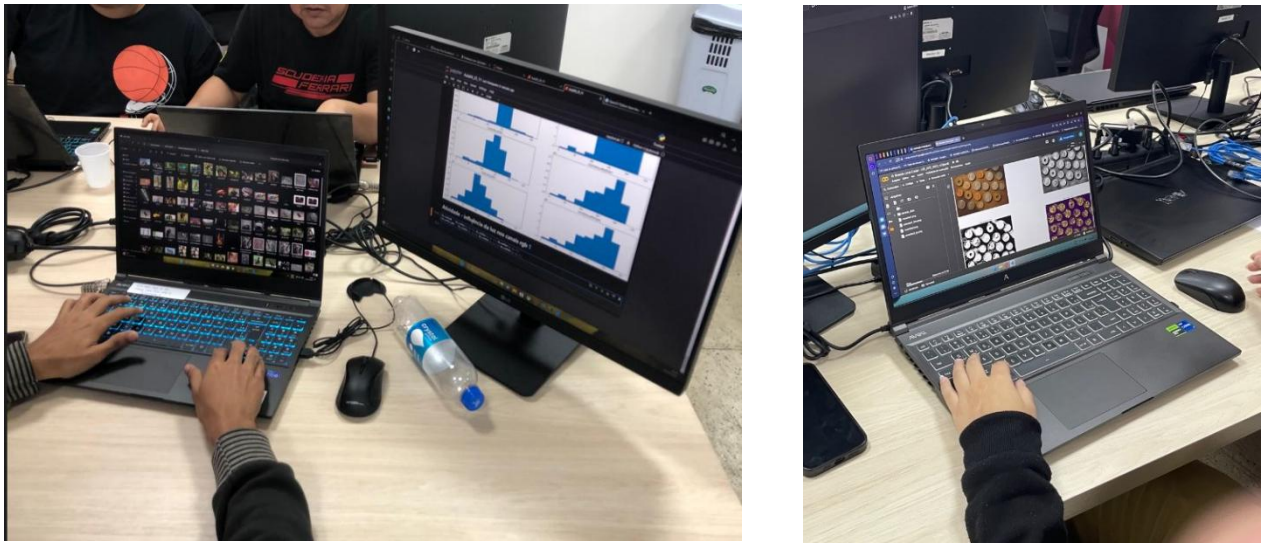


Fonte: Os autores, 2023.

A partir dessa conceituação, foi possível iniciar a codificação utilizando a metodologia *hands-on*. Inicialmente, os alunos geraram diversas imagens e vídeos de objetos e de si mesmos, para aplicar algumas técnicas explicadas e codificadas. Foram utilizados os próprios aparelhos celulares, pois a intenção era realizar variações de luminosidade e deixar que cada aluno tivesse a liberdade para capturar qualquer tipo de objeto presente no laboratório, considerando luzes ligadas e desligadas, assim como uso ou não de iluminação adicional de flash. Inicialmente todas as imagens foram lidas na configuração RGB (Vermelho - *Red,* Verde – *Green,* e Azul - *Blue*).

A partir disso, foram aplicadas técnicas de análise de histograma, como ilustra a Figura 8.2(a), para que os alunos tivessem a percepção da influência da luz na intensidade dos pixels das imagens. Para tal, foi utilizado o método Ravel da biblioteca NumPy (Harris, 2020). Para que essa análise seja possível, é necessário a aplicação deste método, que converte uma matriz multidimensional em uma única dimensão, utilizando a otimização da biblioteca para processar as intensidades e computar as frequências de pixels em cada nível de intensidade, de 0 a 255.

Figura 8.2 - Análise de histograma para intensidade dos *pixels*



(a) (b)

Fonte: Os autores, 2025.

As técnicas de processamento digital de imagens (PDI) formam a base para qualquer solução de visão, sendo cruciais para padronizar a entrada de dados antes da análise. A simples Equalização de Histograma global, por exemplo, muitas vezes falha ao realçar o contraste em regiões menores, resultando em perda de detalhes. Para mitigar isso, utiliza-se a Equalização de Histograma Adaptativa Limitada por Contraste (CLAHE), que aplica a equalização localmente, dividindo a imagem em pequenas células e evitando o excesso de ruído nas regiões homogêneas. Para a tarefa de binarização (transformar a imagem em preto e branco), essencial para segmentação e contagem, a Limiarização de Otsu oferece um método automático e não paramétrico de seleção do limiar muito bom.

Uma vez obtidas, as imagens binarizadas frequentemente contêm ruído indesejado ou pequenas rupturas em objetos. As Operações Morfológicas, baseadas em conjuntos, são

aplicadas para refinar essa segmentação. A combinação de Erosão e Dilatação nas operações de Abertura (Erosão seguida de Dilatação) e Fechamento (Dilatação seguida de Erosão) são ferramentas poderosas. A Abertura é utilizada para remover ruídos pequenos e isolados ("sal e pimenta"), enquanto o Fechamento é empregado para preencher pequenas lacunas nos contornos dos objetos, suavizando o resultado e preparando-o para a extração de características (features).

A correta aplicação dessas técnicas garante que a informação geométrica extraída nas fases subsequentes da visão computacional seja precisa e confiável, independentemente das condições ambientais variáveis. A Figura 8.2(b) mostra um exemplo demonstrado desse pipeline sendo executado.

Todas as atividades práticas foram realizadas com a utilização da linguagem de programação Python 3 (Van Russum & Drake, 2009), pela sua rápida curva de aprendizado, garantindo dessa forma que diversos alunos que não tinham formação em desenvolvimento de *software* conseguissem participar e assimilar sem preocupações com complexidades inerentes à linguagem com maior rigidez sintática. Como alguns alunos ainda não tinham conhecimento da linguagem, foi necessário utilizar cerca de 4 horas do conteúdo para explicar características e realizar codificações utilizando estruturas de dados específicas (listas, tuplas, dicionários), assim como operações de entrada e saída de dados, utilização de estruturas de controle de fluxo e de repetição, sempre reforçando com diversas atividades de fixação. Após esse nivelamento, foi dado início ao conteúdo de processamento de imagens e vídeo, utilizando a biblioteca OpenCV, conforme implementações em Bradski e Kaehler (2008), sendo totalmente de código aberto.

Para entender as demais técnicas de processamento de imagens, foi feita a codificação e explicação de cada uma, com aumento gradual de complexidade e variação de parâmetros, e os alunos replicavam os códigos, aplicando primeiramente em seu conjunto específico de imagens geradas de forma individual, e posteriormente praticando com atividades individuais ou em grupo.

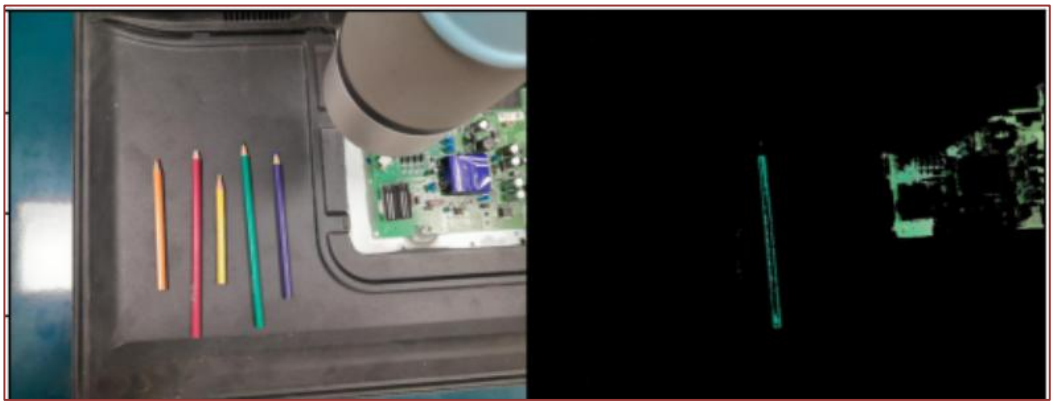
Após o entendimento das questões referentes à aquisição e impacto da luz na geração das imagens, foi dado início ao estudo de técnicas de transformações geométricas importantes para possibilitar ajustes nas imagens, mostrando que se torna desnecessário, em muitos casos, alterar a posição de câmera ou objetos e realizar novas aquisições, a partir de aplicação de transformações como: rotação, redimensionamento – explicando o relevante conceito de interpolação de Jähne (2005). Além disso, operações de mudança de eixos, transposição e espelhamento também foram apresentadas e implementadas. (Szeliski, 2022).

Para a detecção de cores, foi utilizado inicialmente as imagens em RGB, e posteriormente demonstrado que o espaço de cores HSV (*hue, saturation and value*), possibilita maior flexibilidade e precisão no processo de filtro de faixas de cores (Smith,1978).

Com a aplicação dessa técnica, foi possível que os alunos realizassem atividades visando detecção de objetos de cores específicas, conforme exemplo na Figura 8.3, na qual o lado esquerdo mostra a imagem original em RGB, e o lado direito a imagem resultante após definição dos limiares inferior e superior, e aplicação da operação de comparação bit a bit *bitwise_and* conforme Gonzalez & Woods (2010), que possibilita filtrar somente a cor alvo (verde).

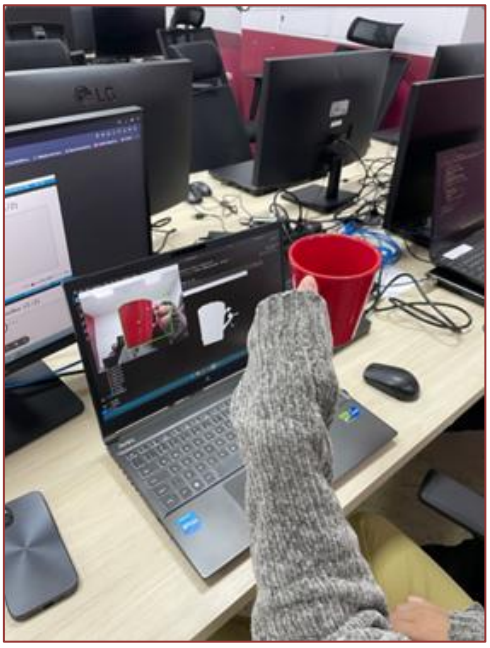
A Figura 8.4 apresenta o resultado do mesmo filtro, utilizado para detectar a cor vermelho em tempo real, conforme o aluno vai movendo a xicara, a máscara vai atualizando os valores e criando um contorno retangular sobre o objeto detectado.

Figura 8.3 – Aplicação de detecção de cores em arquivos de imagem



Fonte: Os autores, 2023.

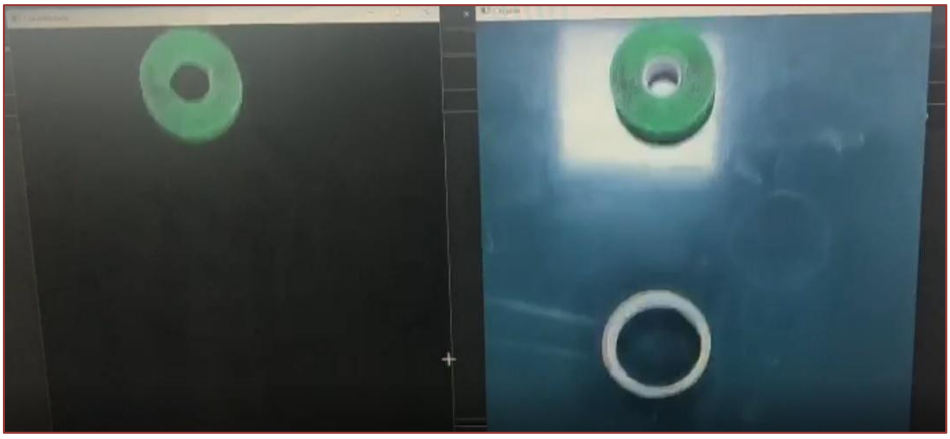
Figura 8.4 – Aplicação de detecção de cores em tempo real



Fonte: Os autores, 2025.

Posteriormente à detecção de cores em imagens, foi feita a explicação de como capturar vídeos em tempo real e realizar a aplicação do filtro de cores, assim como outros filtros. Com utilização da esteira existente, foi então realizada a detecção de objetos em tempo real, conforme a Figura 8.5, que mostra a aplicação da mesma técnica, dessa vez em um vídeo. Do lado esquerdo, a detecção e do lado direito o vídeo original, sem filtro.

Figura 8.5 – Detecção de objetos verdes em tempo real (vídeo)



Fonte: Os autores, 2025.

Ainda referenciando importantes conceitos e técnicas essenciais, foi necessário demonstrar a função filter2D, que realiza o processo de convolução em imagens a partir de um *kernel* específico para cada finalidade (Jäjne, 2005). Os *kernels* podem ser entendidos como filtros, que são matrizes numéricas pequenas, utilizadas para realizar determinadas operações na imagem (matriz) original. Alguns exemplos dessas operações são: suavização; desfoque; nitidez; detecção de bordas, tais como Sobel e LaPlace, ambos demonstrados em Gonzalez & Woods (2010), e Canny (Canny, 1986). Um exemplo da aplicação realizada como atividade prática pode ser visualizada na Figura 8.6, onde do lado esquerdo está a imagem original, e do lado direito a imagem processada com a detecção das bordas.

Figura 8.6 – Aplicação de detecção de borda

Fonte: Os autores, 2025.

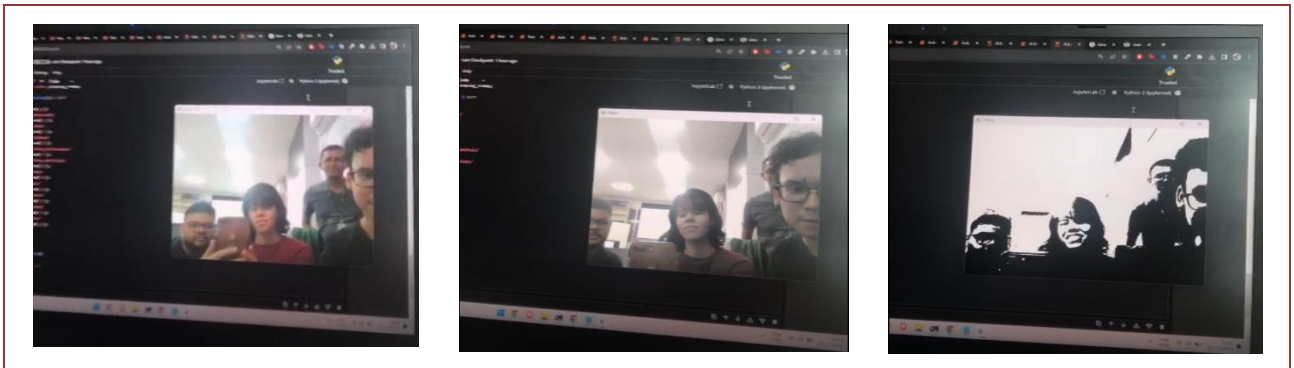
Com o objetivo de possibilitar o entendimento a respeito da necessidade de avaliar a qualidade das imagens, assim como determinados ruídos afetam na aplicação dos filtros, foi utilizada uma imagem de Chowdhury (2017), ilustrada na Figura 8.7, com um nível alto de ruído do tipo *Salt & Pepper*, caracterizado pela presença de pontos (ruídos) brancos (*salt*) e pretos (*pepper*), e a partir disso, foram aplicados filtros de média para observar o resultado. (Gonzales & Woods, 2010). Após a aplicação do filtro, foi possível realizar a diferença entre as imagens e verificar visualmente a quantidade de ruído eliminado da imagem original.

Figura 8.7 – Aplicação e visualização da quantidade de ruído existente na imagem

Fonte: Adaptado a partir de Chowdhury, 2017.

Para finalizar os tópicos relativos a esta seção, foi realizada uma atividade na qual os alunos criaram funções para capturar vídeos em tempo real pela câmera, e a partir de teclas configuradas, determinados filtros pré-definidos foram aplicados, conforme a Figura 8.8, que mostra a aplicação de conversão de formato dos vídeos de RGB para tons de cinza e binário.

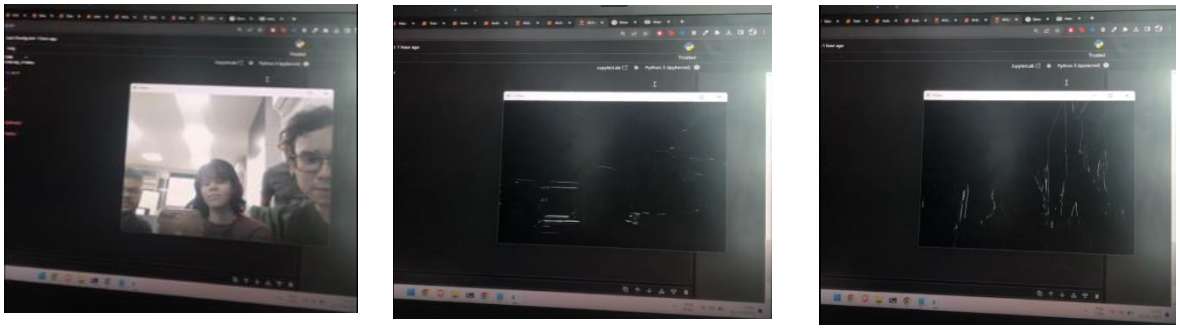
Figura 8.8 – Conversão de vídeos RGB em tons de cinza e binário, em tempo real



Fonte: Os autores, 2023.

Além dessas conversões de imagens, outros filtros aplicados também em tempo real foram os de detecção de linhas horizontais e de linhas verticais (Filtro de Sobel XY) em imagens em tons de cinza, conforme a Figura 8.9.

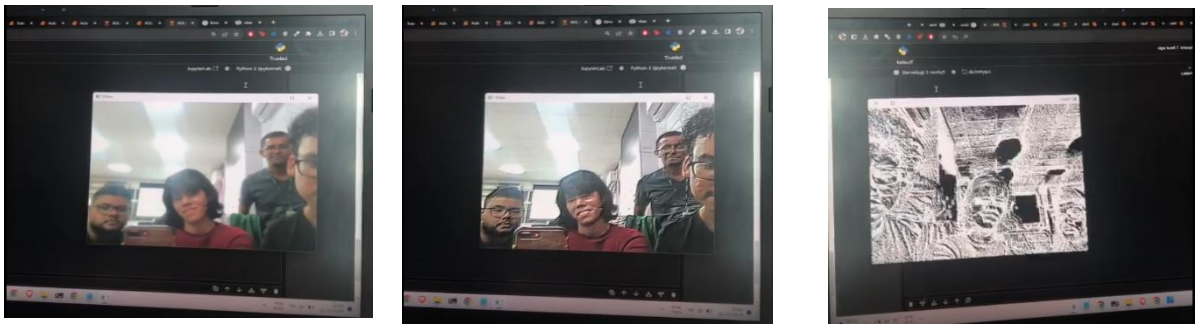
Figura 8.9 – Detecção de linhas horizontais e verticais em tempo real



Fonte: Os autores, 2023.

Por fim, na Figura 8.10 é mostrado o resultado da aplicação dos filtros de realce (sharpen), que realça as diferenças de intensidades entre os pixels adjacentes; de relevo (emboss), responsável por destacar as bordas e criar um efeito que imita uma superfície elevada; e de Sobel, que realça as bordas detectadas, em tons de cinza.

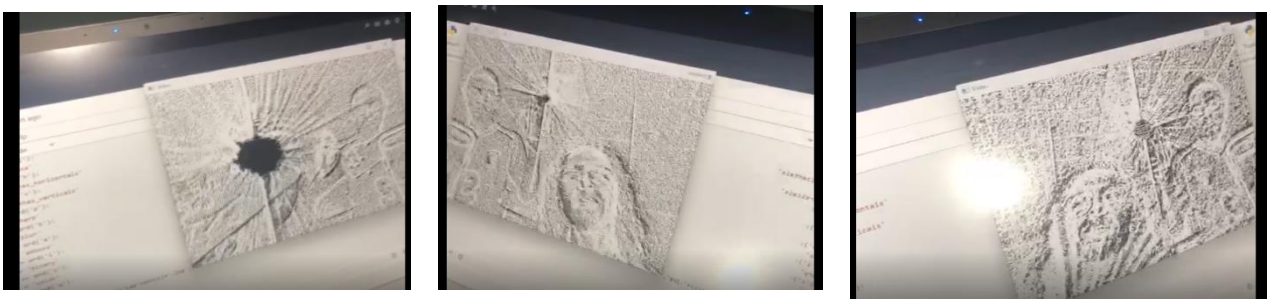
Figura 8.10 – Aplicação dos filtros sharp, relevo e Sobel em tempo real



Fonte: Os autores, 2023.

Esta atividade despertou a curiosidade de vários alunos, possibilitando que se aprofundassem com a realização de diversos outros testes, já que as capturas estavam sendo feitas em tempo real. Isso permitiu que, por exemplo, fossem feitas análises de como determinados filtros, como o Sobel, se comportam em quadros capturados com o auxílio de luzes do tipo *flash*. Esses resultados são ilustrados na Figura 8.11, mostrando os feixes da frequência luminosa e com o foco principal da luz invertido.

Figura 8.11 – Resultado de uso de flashes em tempo real com o método Sobel



Fonte: Os autores, 2023.

8.2.4. APLICAÇÕES EM VISÃO COMPUTACIONAL

A Visão Computacional (VC) representa a etapa em que o computador, após o Processamento Digital de Imagens (PDI), passa a interpretar o conteúdo da cena. Enquanto o PDI foca em operações de sinal para realce, filtragem e segmentação (GONZALEZ; WOODS, 2018), a VC integra geometria, calibração e técnicas de reconhecimento de padrões para tomar decisões e quantificar informações. O foco na indústria é transformar a percepção visual em dados metrológicos e de rastreabilidade para o controle de qualidade e a orientação robótica.

8.2.4.1. O PAPEL CRÍTICO DO PRÉ-PROCESSAMENTO ROBUSTO

Em ambientes fabris, a variabilidade na iluminação, contraste e ruído é constante, exigindo que a etapa de pré-processamento seja altamente robusta.

O uso de técnicas avançadas de PDI é fundamental para estabilizar a imagem antes de qualquer análise geométrica ou de feature. Por exemplo, a Equalização de Histograma Adaptativa Limitada por Contraste (CLAHE) é preferida à equalização global, pois realça o contraste localmente, preservando detalhes importantes. Após a segmentação (por exemplo, via Limiarização de Otsu para determinação automática do threshold), o refinamento dos contornos é feito através de Operações Morfológicas, como a Abertura e o Fechamento. Essas operações removem ruídos isolados e preenchem pequenas rupturas, respectivamente, garantindo que os objetos a serem analisados apresentem formas geométricas limpas. A aplicação coerente dessas técnicas em um pipeline de inspeção é crucial para o sucesso da aplicação, como pôde ser visto na Figura 8.3.

8.2.4.2. CALIBRAÇÃO E METROLOGIA

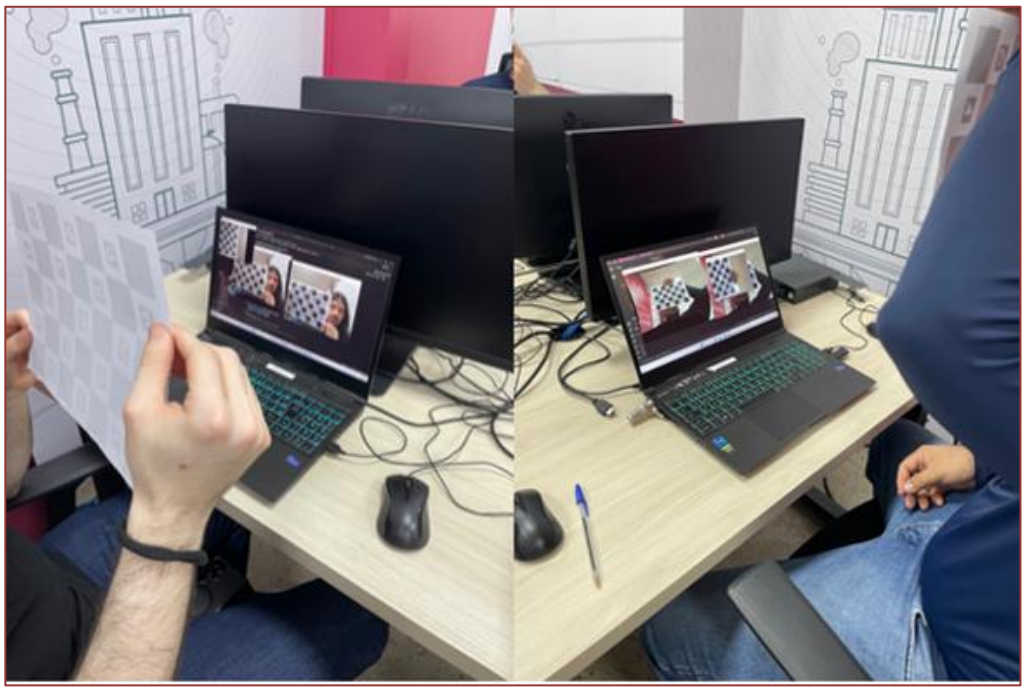
Na indústria, a capacidade de medir com precisão a partir de uma imagem é crucial. Uma imagem digital é uma representação 2D de um mundo 3D, e esta transformação introduz distorções significativas, causadas principalmente pela ótica das lentes (distorção radial e tangencial) e pela perspectiva.

O processo de Calibração de Câmeras é o passo fundamental da metrologia por visão. Ele determina os parâmetros intrínsecos (características da câmera, como distância focal e centro ótico) e extrínsecos (posição e orientação da câmera no espaço 3D),

permitindo corrigir geometricamente a imagem e mapear pixels para coordenadas métricas reais.

Em ambientes práticos, o uso de padrões fiduciais como as etiquetas ChArUco (uma combinação de marcadores ArUco - padrões binários quadrados fáceis de detectar - com tabuleiros de xadrez) oferece uma solução mais robusta para o processo de calibração do que os tabuleiros de xadrez puros, facilitando a detecção dos cantos mesmo em imagens de menor qualidade. É um processo detalhado, essencial para a metrologia industrial, como é visto na Figura 8.12. Uma calibração bem-sucedida garante que a solução de visão computacional possa ser utilizada para medições precisas e para tarefas de alinhamento e pose em tempo real (Forsyth; Ponce, 2015).

Figura 8.12 – Processo de Calibração de Câmeras utilizando o padrão ChArUco (Combinação de Chessboard e Marcadores ArUco), evidenciando a correção de distorção (undistort)



Fonte: Os autores, 2025.

8.2.4.3. EXTRAÇÃO DE FEATURES INVARIANTES E RECONHECIMENTO DE PADRÕES

O reconhecimento de objetos e padrões em ambientes industriais exige algoritmos que sejam imunes a variações de escala, rotação e iluminação. Para isso, a VC clássica utiliza Descritores Locais (Local Features), que são pontos únicos e distintivos em uma imagem.

Descritores como SIFT (Scale-Invariant Feature Transform) e ORB (Oriented FAST and Rotated BRIEF) analisam a vizinhança de um ponto de interesse e criam um vetor numérico (o descritor) que é invariante à rotação e escala da imagem. Essa invariância permite que um objeto ou um logo de referência seja reconhecido com sucesso, mesmo

que seja visto de diferentes ângulos ou distâncias na linha de produção (Solem, 2012). A Figura 8.13 mostra o uso do SIFT para reconhecer padrões.

Figura 8.13 – Exemplo de extração de Descritores Locais com o algoritmo SIFT, demonstrando a identificação de keypoints e o matching entre uma imagem de referência e a imagem da cena

Fonte: Os autores, 2025.

O pipeline de reconhecimento geralmente envolve:

- Extração de Features: Identificação de keypoints e computação de seus descritores (como visto nas práticas de SIFT/ORB);
- Matching: Comparação dos descritores da imagem de entrada com os descritores de uma imagem de referência, utilizando algoritmos como KNN (K-Nearest Neighbors);
- Verificação Geométrica: Aplicação do algoritmo RANSAC (Random Sample Consensus) para estimar a Homografia (transformação geométrica) que alinha a imagem de referência com a imagem atual, filtrando matches errados e garantindo a correta verificação geométrica do objeto.

Essas técnicas são a espinha dorsal de sistemas que realizam a verificação de *Template Matching* para alinhamento e inspeção de presença/ausência de componentes, garantindo que a orientação e a posição das peças estão corretas antes da próxima etapa robótica.

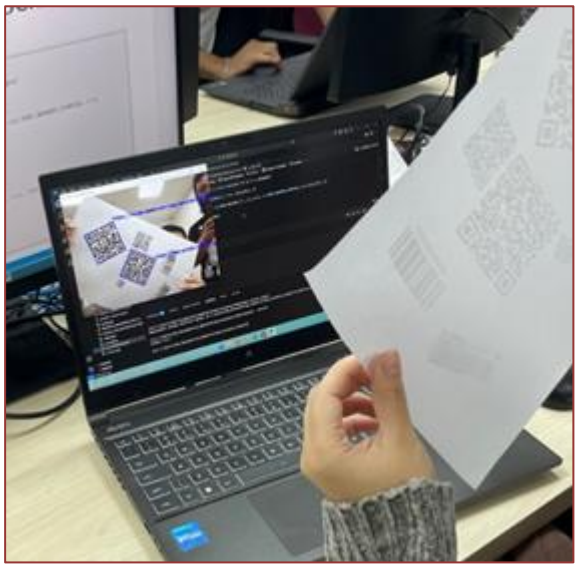
8.2.4.4. RASTREABILIDADE: OCR E LEITURA DE CÓDIGOS

A rastreabilidade é uma das aplicações mais diretas e de alto valor da VC. A disciplina explorou as ferramentas para detecção e decodificação de padrões de identificação:

- Códigos Unidimensionais (1D): Como códigos de barras;
- Códigos Bidimensionais (2D): Incluindo QR Codes e Datamatrix;
- Optical Character Recognition (OCR): Para a leitura de texto em rótulos e etiquetas.

A integração de técnicas de Processamento de Imagens (como a Limiarização de Otsu e filtros de realce) na fase de pré-processamento é crucial para isolar e melhorar a qualidade do código antes de sua decodificação, garantindo alta taxa de sucesso de leitura, mesmo em embalagens danificadas ou com baixo contraste. O resultado prático desta decodificação pode ser observado na Figura 8.14.

Figura 8.14 - Detecção e decodificação de texto via OCR em tempo real



Fonte: Os autores, 2025.

8.2.5. APRENDIZADO DE MÁQUINA E APLICAÇÕES EM VISÃO COMPUTACIONAL

O aprendizado de máquina constitui o estado da arte em problemas de visão computacional com objetivo de detecção, classificação e reconhecimento de objetos. Por este motivo, conhecer seus fundamentos, assim como quais são as etapas necessárias para utilizar modelos existentes é crucial para qualquer projeto que envolva visão computacional. A codificação de tais modelos é realizada de forma bem diferente da programação tradicional, onde são apresentados os dados de entrada e elaborados os programas com determinada logica para que seja gerada a saída (resultado). Em aprendizado de máquina, são definidos os dados de entrada e a saída esperada (resultado), e o modelo realiza as inferências necessárias para estabelecer funções de aproximação, “aprendendo” características dos dados. Estes modelos podem ser usados para dados de texto ou imagens.

Em tarefas envolvendo imagens, os modelos mais robustos e precisos utilizam várias camadas dispostas sequencialmente entre a camada de entrada e de saída, e são conhecidos como modelos de *deep learning.* Os modelos utilizados durante as aulas foram basicamente modelos supervisionados de classificação, conforme James *et al*. (2013), e esta seção trata de conceitos sobre esse tipo de modelo, e como ele foi aplicado durante as aulas.

O objetivo de um modelo de classificação é prever a categoria a qual uma instância (amostra) pertence, com base em suas características (Duda; Hart; Stork, 2001). O resultado de uma classificação é um conjunto de valores percentuais, indicando probabilidades para as classes para as quais o modelo foi treinado.

Treinar um modelo significa realizar extração das características mais significativas dos dados, e a partir disso, ao submeter uma imagem qualquer, o modelo será capaz de dizer qual classe a imagem pertence. Modelos de classificação podem ser projetados para identificar objetos de *n* classes distintas. (Alpaydin, 2020).

No caso de modelos para tarefas em imagens, as atividades realizadas durante o treinamento são normalmente baseadas na aplicação de filtros de convolução, ou Redes Neurais Convolucionais (CNNs) (Lecun, Bengio, Hinton, 2015).

O processo para utilização de um modelo de classificação pode ser entendido em duas etapas: a primeira é o treinamento do modelo, e a segunda ocorre quando o modelo já foi treinado e pode ser utilizado para inferências, com a submissão de entradas para a classificação. Para realizar treinamento de modelos de *deep learning* é necessário utilizar uma quantidade extremamente grande de imagens para que as características relevantes dos dados sejam extraídas e associadas. Uma outra questão a ser observada é a disponibilidade de grandes conjuntos de dados públicos para possibilitar o treinamento. O processo de construção desses conjuntos de dados requer um investimento de tempo e recurso computacionais nem sempre disponíveis. Por essas questões, opta-se normalmente pela utilização de modelos já treinados em grandes bases de dados.

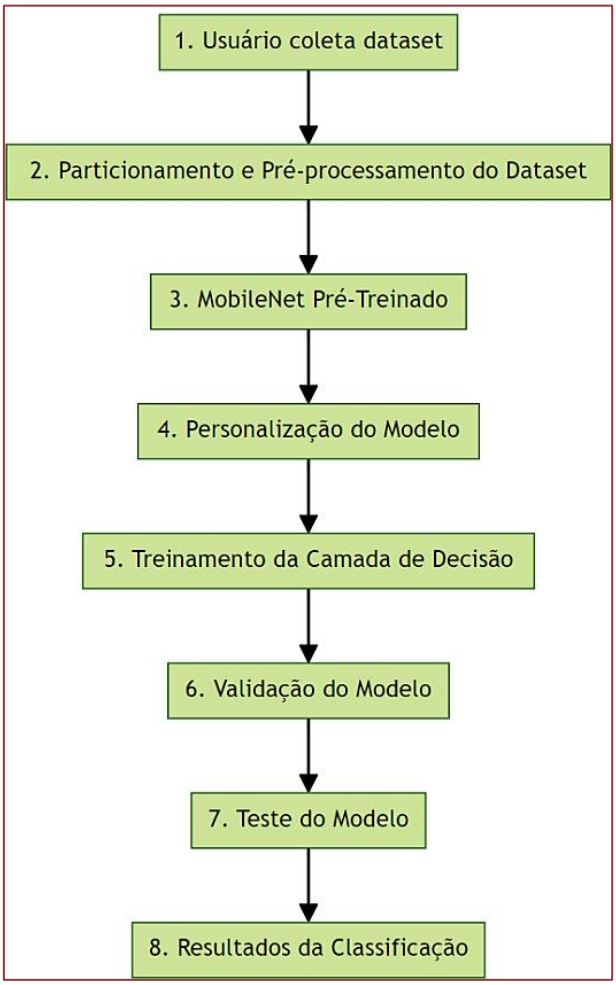
Um dos primeiros conjuntos de dados disponibilizado foi o ImageNet, por Deng *et al*. (2009), e a partir dele, diversos modelos foram treinados com as milhões de imagens que o conjunto contém, com 1000 classes disponíveis.

Shin *et al*. (2016) utilizaram modelos "pré-treinados", ou seja, substituíram a camada que realiza a classificação das imagens (camada de decisão), pois todas as demais camadas são capazes de identificar as características básicas utilizadas no processo de identificação das imagens, pois os neurônios existentes já contêm os pesos (valores) ajustados para o reconhecimento. Essa técnica é conhecida como *Transfer Learning*, utilizada em grande parte dos projetos que utilizam modelos de *deep learning* e por isso, optamos por utilizar nas aulas alguns modelos pré-treinados (Kornblith, Shlens, Le. 2019).

As primeiras atividades foram utilizando o conjunto de dados CIFAR-10, elaborada por Krizhevsky, Nair e Hinton (2009), para que os alunos entendessem como estruturar e configurar parâmetros de um modelo e realizar o carregamento do modelo pré-treinado. Em seguida, foram realizadas atividades práticas com mais 3 modelos: ResNet, uma rede mais profunda, com mais camadas entre as camadas de entrada e de saída, He et al. (2016); MobileNet, um modelo mais otimizado para hardware com menor poder de processamento, Howard et al. (2017) e EfficientNet, um modelo que exige menos recursos e que obtém melhor desempenho. (Tan & Le, 2019).

A Figura 8.15 ilustra resumidamente as etapas necessárias para utilização de um modelo pré-treinado na MobileNet: aquisição de imagens-alvo; definição e particionamento do modelo; pré-processamento das imagens coletadas; carregamento dos pesos das camadas; personalização, com a troca da camada de decisão pela camada que representa as classes-alvo do problema em questão; treinamento da camada de decisão; validação e teste do modelo; e finalmente, avaliação dos resultados (desempenho) do modelo.

Figura 8.15 – Etapas para utilização do modelo pré-treinado MobileNet



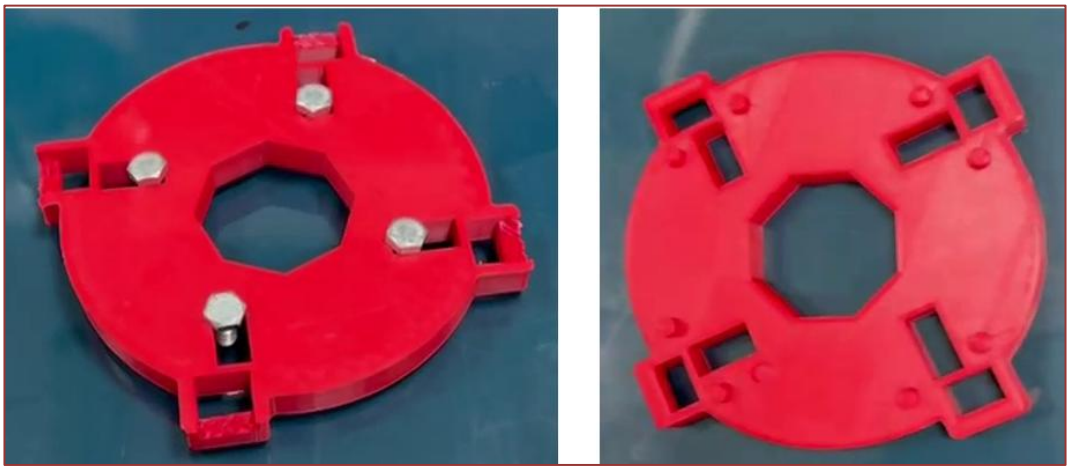
Fonte: Os autores, 2023.

O funcionamento básico pode ser entendido conforme a sequência de cada numeração da legenda, e todos os passos foram seguidos nas atividades práticas:

1) O usuário realiza a aquisição ou coleta das imagens e/ou vídeos necessários, de acordo com o tipo de problema que precisa ser aplicado. No caso de problemas de visão computacional, os dados de entrada são as imagens ou vídeos, que são agrupados por classe em diretórios diferentes, e as saídas são as classes dos objetos (nomes dos diretórios). Como exemplo de aplicação de treinamento, podemos utilizar imagens de produtos que precisam ser analisados em relação à ocorrência ou não de defeitos de fabricação. Para que isso seja possível, foi

necessário fazer a geração imagens de produtos normais (não defeituosos) e outro conjunto de imagens dos produtos defeituosos. Como analogia, podemos exemplificar que em determinada linha de produção, uma peça precisa ter exatamente 4 furos. Ao gerar imagens desse tipo, podemos utilizar a estratégia de nomear o diretório das imagens com 4 furos como “Aprovado”, e todas as outras imagens de produtos defeituosos, com 1, 2 ou 3 furos serão agrupadas em outro diretório, com o nome “Defeito”. É importante que o número de imagens seja equivalente entre as classes, e com isso, criamos um dataset balanceado e evitamos que o modelo gere alguma tendência por “enxergar” mais características de determinada classe. Se quisermos somente que o modelo classifique imagens como produto aprovado ou defeituoso, podemos entender que esse é um problema de classificação binária. (Konrad, Lohmann, Abel, 2019). Para realizar uma atividade prática simulando os projetos que cada equipe precisaria atuar para resolver, foi orientado que cada uma procurasse peças que pudessem ser utilizadas para simular problemas similares. A Figura 8.16 ilustra uma peça utilizada por uma das equipes em seu projeto na jornada hands on;

Figura 8.16 - Peças utilizadas para simular itens de um dos projetos da jornada *hands on*



Fonte: Os autores, 2025.

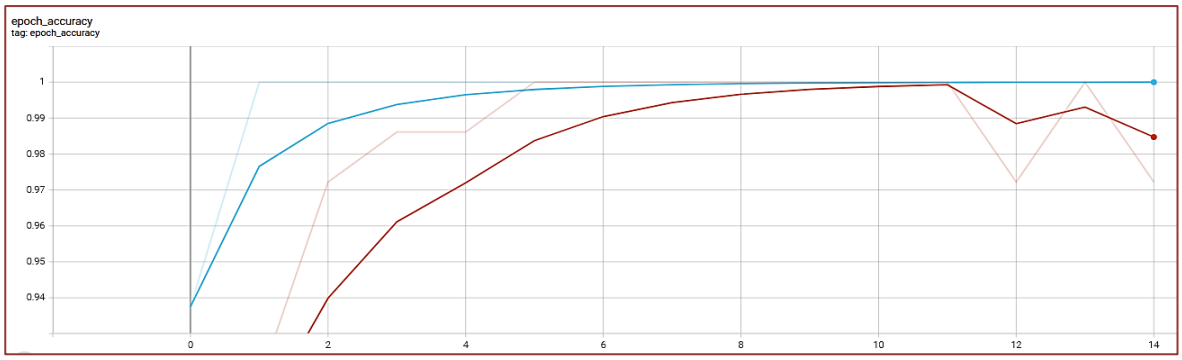
2) O próximo passo é o particionamento do dataset, ou seja, divisão de um número determinado de imagens, tanto para a classe “Aprovado” quanto para a classe “Defeito”. Esse processo é feito configurando parâmetros entre “treino”, “validação” e “teste”, e optamos pela proporção percentual de 70/10/20, respectivamente. Após o particionamento, é feito o pré-processamento do dataset, ou seja, são aplicadas técnicas de acordo com as características do modelo escolhido. No mínimo é feito o redimensionamento das imagens para que possam ser entendidas pelo modelo, pois no caso do MobileNet, a resolução das imagens para entrada no modelo deve ser (224, 224);
3) É feita então a definição do modelo a ser treinado, com o carregamento do modelo pré-treinado, utilizando todas as camadas a não ser a de decisão “congeladas”, evitando dessa forma que os seus pesos sejam atualizados;

4) A personalização do modelo pode ser realizada com acréscimos de camadas adicionais e aplicação de várias técnicas que ajudem em melhores resultados. Além disso, podemos considerar realização de ajustes de hiper parâmetros, tais como número de épocas e tamanho do lote;

5) A partir das etapas de definição e personalização, inicia-se o treinamento do modelo. Em do MobileNet tentar reconhecer as 1000 classes para as quais o modelo ImageNet treinado inicialmente, irá utilizar nossa classificação binária de acordo com a estrutura dos diretórios, , com a substituição da camada de classificação;

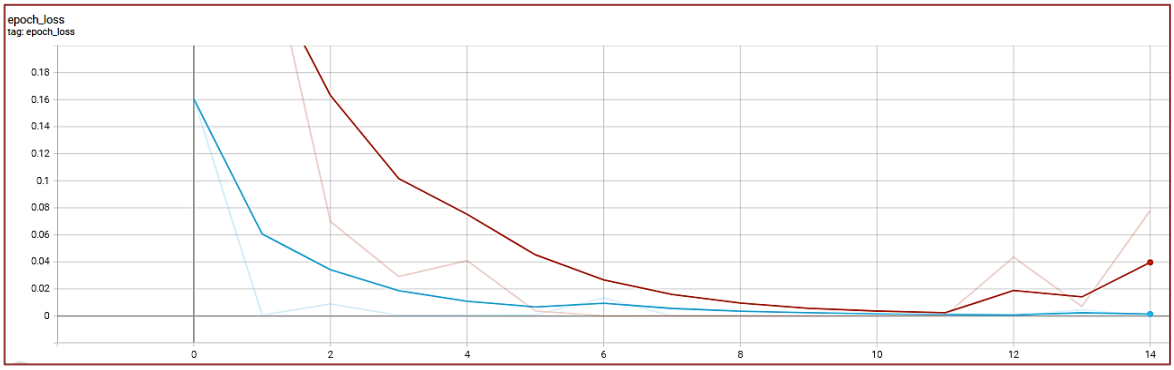
6) Para modelos de classificação, normalmente é utilizado como indicador de desempenho a métrica acurácia, e é feita a análise da função de perda, ou erro do modelo, tanto nas etapas de treinamento, quanto na validação e teste. A acurácia é definida pela proporção (%) de previsões corretas resultante do treinamento e validação, ou seja, qual a taxa de acerto da classificação. A Figura 8.17 ilustra o comportamento isolado da métrica "acurácia", para 15 épocas que o modelo foi treinado, com a utilização do Tensorboard, bastante utilizado para esta finalidade, como exemplificado nos trabalhos de Pranto e Faruk (2025) e Saadi et al. (2025). representa o acompanhamento do treinamento por 10 épocas do modelo, através do qual os alunos puderam entender o processo de diminuição do erro e aumento da acurácia ao longo das épocas do treinamento, tanto em treino quando em validação. Outra variável que precisa ser analisada é o erro (loss). A Figura 8.18 exibe como se comportou essa variável durante o treinamento das 15 épocas, demonstrando um possível overfitting a partir da época 11;

Figura 8.17 - Evolução da acurácia por época no treinamento (vermelho) e validação (azul) do modelo pré-treinado da MobileNet



Fonte: Os autores, 2023.

Figura 8.18 - Evolução do erro por época no treinamento (vermelho) e validação (azul) do modelo pré-treinado da MobileNet



Fonte: Os autores, 2023.

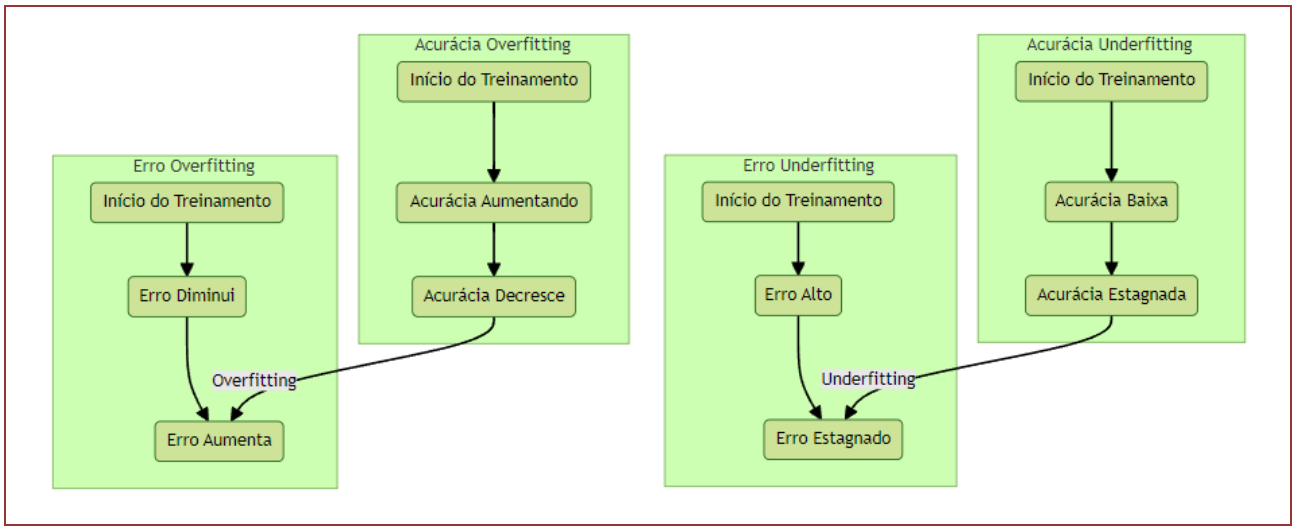
1. Para realizar os testes no modelo, são feitas várias inferências com imagens que o modelo não conhece, ou seja, não foram utilizadas no treino ou na validação. Conforme a proporção inicialmente definida, 20% das imagens foram destinadas para o teste.
2. Por fim, é aferida a acurácia e erro no teste.

Todas as etapas descritas foram codificadas e testadas utilizando o framework Keras, criada por Chollet (2015), com aplicação de diversas variações de otimizadores, conforme Goodfellow, Bengio e Courville (2016) e Gulli e Pal (2017), para que os alunos tivessem a noção do impacto dos otimizadores no treinamento e resultados no desempenho do modelo.

Além disso, dependendo da quantidade de imagens que os alunos tinham gerado, foi necessário explicar questões relativas a *overfitting* e *underfitting*, pois com quantidades insuficientes de imagens coletadas, o modelo não conseguiu se adaptar ao novo conjunto de dados. A Figura 8.19 ilustra o tipo de *insight* demonstrado para os alunos, para que pudessem perceber a ocorrência de *overfitting* (lado esquerdo) e *underfitting* (lado direito), analisando valores de acurácia e erro durante o processo de treinamento.

Para contornar esses problemas, foi demonstrado como realizar o ajuste do modelo, descongelando mais camadas além da camada de decisão para o treino, e também, explicação e demonstração da técnica de aumento de dados (data augmentation), conforme Nusrat e Jang (2018) e Terras et al. (2025), com operações de rotação com limite de até 30 graus, espelhamento, zoom, variações de brilho, inserção de ruído, deslocamento na altura e deslocamento na largura. As técnicas de aumento de dados têm demonstrado eficiência mesmo utilizando modelos tradicionais como demonstrado em diversas pesquisas, como na de Liu et al. (2024), gerando variações das imagens a partir do ruído laplaciano e gaussiano após converter as imagens para tons de cinza.

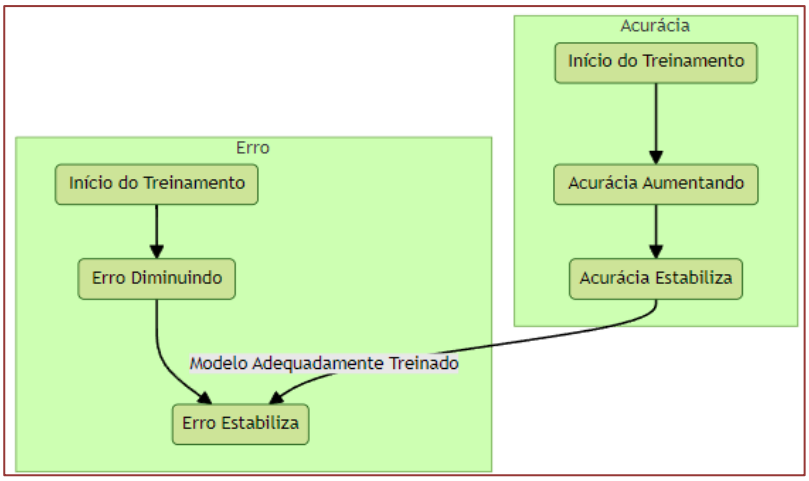
Figura 8.19 – Indicação de uma técnica de análise de valores de erro e de acurácia, para detectar *overfitting* e *underfitting* durante o treinamento de modelos



Fonte: Os autores, 2023.

Foi explicado também como seria o comportamento ideal de um modelo, sem a ocorrência dos problemas mencionados. A Figura 8.20 ilustra esse comportamento, ainda com a verificação da acurácia e do erro durante as épocas do treinamento.

Figura 8.20 – Comportamento ideal da acurácia e erro durante o treinamento de modelos

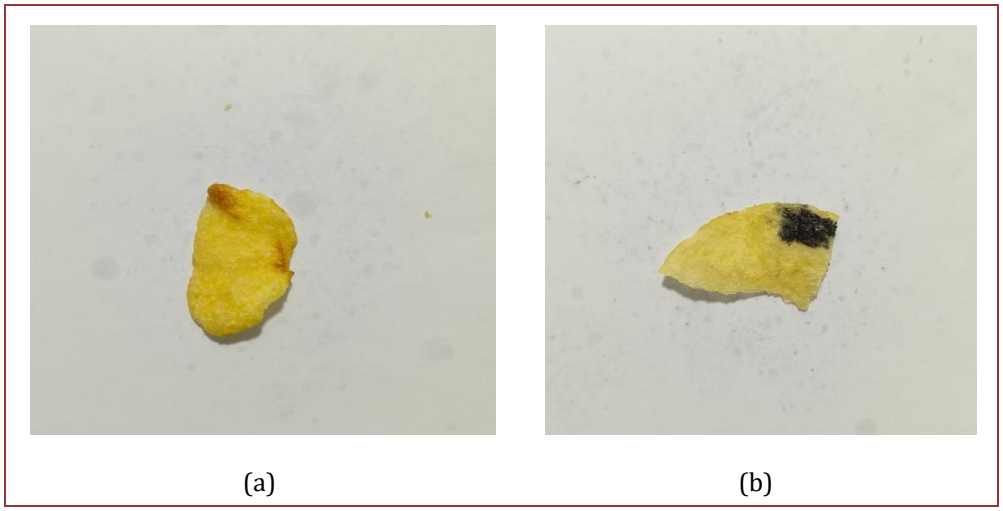


Fonte: Os autores, 2023.

Para possibilitar a análise detalhada de um caso real, um conjunto de dados, chamado “Potato Chips” usado em Pedreschi et al. (2016) e Taspinar e Koklu (2024) passou a ser utilizado como problema de classificação. Além de todos os passos anteriores, os alunos também entenderam a analisar a matriz de confusão e relatório de classificação, que permitem entender de forma mais detalhada e por classe os erros

ocorridos. As imagens consistem em batatas aprovadas ou reprovadas no controle de qualidade de uma fábrica de chips de batata, conforme ilustradas na Figura 8.21(a) – caso positivo, sem defeito, e a Figura 8.21(b) – caso negativo, defeituosa.

Figura 8.21 – Imagens das duas classes do dataset "Potato Chips": (a) Caso positivo, ou chip sem defeito, e (b) Caso negativo, ou chip defeituoso



(a) (b)

Fonte: Os Autores (2025).

A Figura 8.22 ilustra a matriz de confusão seguida do relatório de classificação após o treinamento do conjunto de dados "Potato Chips", com treinamento no MobileNet. Pode-se verificar que para a classe negativa ("Defective"), o modelo classificou com precisão de 100% e teve revocação (recall) de 98%, e com escore f1 equilibrado. No caso da classe positiva ("Non Defective"), o modelo deu mais ênfase para a revocação. Pode-se considerar um bom resultado, com poucas épocas e utilizando transfer learning a partir do modelo pré-treinado do conjunto de dados "ImageNet".

Figura 8.22 - Resultados do modelo com a matriz de confusão e relatório de classificação para o problema de classificação binária apresentado com o conjunto de dados "Potato Chips"

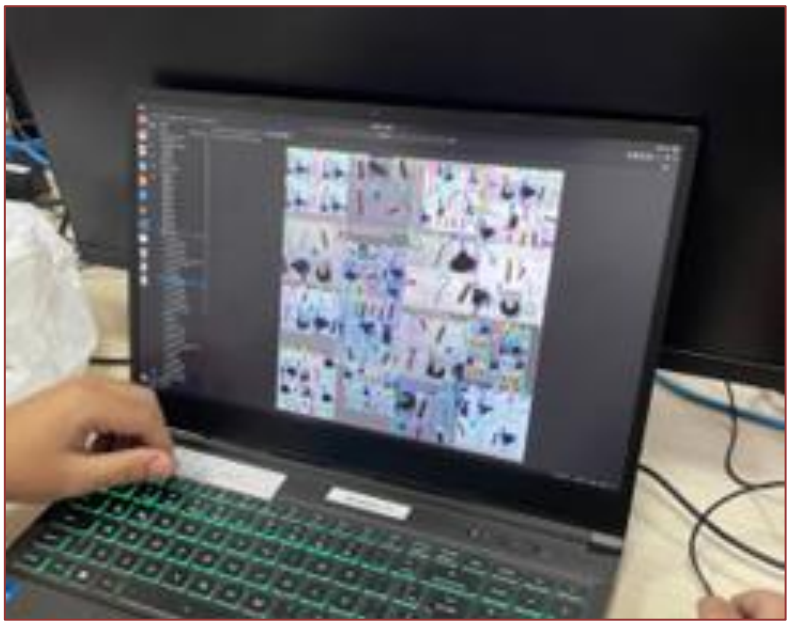
```
Confusion Matrix:
[[ 90   2]
 [  0 100]]

Classification Report:
              precision    recall  f1-score   support

   Defective       1.00      0.98      0.99        92
Not Defective      0.98      1.00      0.99       100

    accuracy                           0.99       192
   macro avg       0.99      0.99      0.99       192
weighted avg       0.99      0.99      0.99       192
```

Fonte: Os autores, 2025.

Embora a classificação (como vista no exemplo anterior das batatas) determine se uma peça está aprovada ou reprovada, a automação industrial frequentemente requer a identificação precisa da localização do defeito ou do componente para a pega robótica. Para suprir essa necessidade, o curso avançou para a Detecção de Objetos.

Arquiteturas como YoLo (You Only Look Once) foram apresentadas como o estado da arte industrial, pois são capazes de realizar a detecção e classificação em uma única passada pela rede, resultando em velocidades de inferência próximas ao tempo real. Essas redes permitiram criar Provas de Conceito (PoCs) eficazes para tarefas onde era necessário desenhar caixas delimitadoras (bounding boxes) ao redor dos defeitos, como ilustrado na Figura 8.23.

Figura 8.23 - Detecção multi-classe de vários objetos com a arquitetura YoLo, mostrando as caixas delimitadoras e a classificação das classes (e.g., "parafuso", "porca", “etiqueta”, etc)

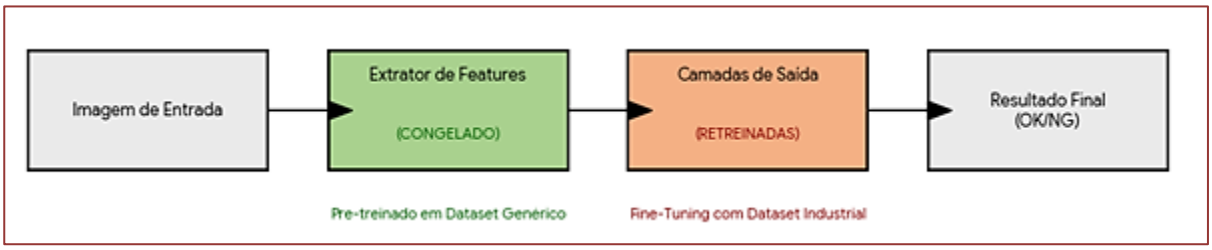


Fonte: Os autores, 2025.

Para solidificar o entendimento do método de Transfer Learning utilizado nas práticas com MobileNet e ResNet, foi apresentado um fluxograma que resume a estratégia de aproveitar redes profundas em cenários com escassez de dados (comum na indústria).

A estratégia consiste em manter as camadas iniciais da rede (que detectam bordas e texturas genéricas) congeladas, retreinando apenas as camadas finais com o dataset específico do chão de fábrica. Este método minimiza a necessidade de dados e acelera drasticamente o treinamento, conforme detalhado visualmente na Figura 8.24.

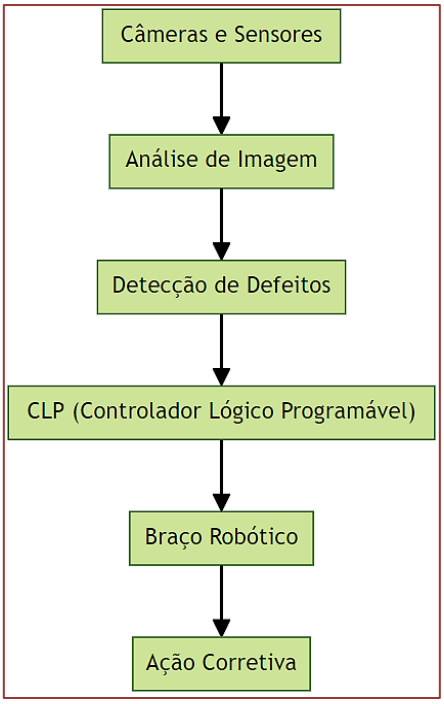
Figura 8.24 - Fluxograma do Transfer Learning em CNNs: Camadas iniciais (Extração de Features) são congeladas, e as camadas finais (Classificação) são retreinadas com um dataset industrial específico



Fonte: Os autores, 2025.

Com as inúmeras práticas realizadas no decorrer da disciplina, foi possível que os alunos entendessem a importância da Visão Computacional e diversas aplicações na área da indústria. Integrada à robótica industrial, e conforme Mane (2021), existem inúmeros avanços tecnológicos, principalmente com a aplicação de modelos de deep learning, sendo fundamental para a detecção de defeitos e otimização de processos produtivos. A Figura 8.25 apresenta, de forma resumida, como é a interação desde a aquisição feita por câmeras ou sensores, análise das imagens, detecção de defeitos, comunicação com braço robótico via CLP, e a ação corretiva, que pode ser um programa previamente configurado, com por exemplo, retirar a peça defeituosa da esteira para uma área específica.

Figura 8.25 - Etapas integradas da aplicação de Visão Computacional integrada à robótica



Fonte: Os autores, 2023.

A última etapa da disciplina foi para acompanhar os projetos de cada equipe, e ainda, para orientar como os alunos poderiam utilizar dois computadores remotos com GPUs mais adequadas para treinamentos do que os notebooks utilizados durante as aulas. Os computadores possuíam 12 GB de NVRAM, e foram previamente instalados e configurados com todos os drivers compatíveis. A partir disso, um integrante de cada time ficou responsável para fazer upload dos códigos python e conjuntos de dados, a partir do FileZilla, e a partir de uma conexão ssh e usando o aplicativo task spooler poderiam deixar seus Jobs executando em uma fila, tendo o cuidado de gerar todos os logs necessários para análise do treinamento e dos resultados de forma posterior.

REFERÊNCIAS

[1] ALPAYDIN, Ethem. **Introduction to machine learning**. MIT press, 2020.

[2] BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV: Computer vision with the OpenCV library**. " O'Reilly Media, Inc.", 2008.

[3] CANNY, John. A computational approach to edge detection. **IEEE Transactions on pattern analysis and machine intelligence**, n. 6, p. 679-698, 1986.

[4] CHOLLET, F. (2015). Keras. [Software]. Disponível em https://keras.io.

[5] CHOWDHURY, Debkumar et al. A quantum study on digital image noises and their in-depth clusterization. In: **2017 4th International Conference on Opto-Electronics and Applied Optics (Optronix)**. IEEE, 2017. p. 1-7.

[6] DENG, Jia et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE conference on computer vision and pattern recognition**. Ieee, 2009. p. 248-255.

[7] DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. 2.ed. New York: John Wiley & Sons, 2001.

[8] FORSYTH, D. A.; PONCE, J. **Computer Vision**: A Modern Approach. 2.ed. Upper Saddle River, NJ: Pearson Education, 2015.

[9] GONZALEZ, Rafael C.; WOODS, Richard E. Processamento digital de imagem. **Pearson, ISBN-10: 8576054019**, v. 10, 2010.

[10] _____. **Digital Image Processing**. 4.ed. New York: Pearson, 2018.

[11] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. MIT press, 2016.

[12] GULLI, Antonio; PAL, Sujit. Deep learning with Keras. Packt Publishing Ltd, 2017.

[13] HARRIS, Charles R. et al. Array programming with NumPy. **Nature**, v. 585, n. 7825, p. 357-362, 2020.

[14] HE, Kaiming et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016. p. 770-778.

[15] HOWARD, Andrew G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.

[16] JÄHNE, Bernd. **Digital image processing**. Springer Science & Business Media, 2005.

[17] JAMES, Gareth et al. **An introduction to statistical learning**. New York: springer, 2013.

[18] KRIZHEVSKY, A., NAIR, V., & HINTON, G. (2009). CIFAR-10 (Canadian Institute for Advanced Research) [Dataset]. Disponível em: https://www.cs.toronto.edu/~kriz/cifar.html

[19] KORNBLITH, Simon; SHLENS, Jonathon; LE, Quoc V. Do better imagenet models transfer better?. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. 2019. p. 2661-2671.

[20] KONRAD, Thomas; LOHMANN, Lutz; ABEL, Dirk. Surface defect detection for automated inspection systems using convolutional neural networks. In: **2019 27th Mediterranean Conference on Control and Automation (MED)**. IEEE, 2019. p. 75-80.

[21] LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, v. 521, n. 7553, p. 436-444, 2015.

[22] LIU, Hengxu et al. Surface defect detection based on machine learning and data augmentation. In: **2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology** (ICCASIT). IEEE, 2024. p. 1098-1103.

[23] MANE, Tanmay et al. Identification and Classification of Industrial Elements using Artificial Intelligence and Image Processing Techniques. In: **2021 International Conference on Emerging** Smart Computing and Informatics (ESCI). IEEE, 2021. p. 165-169.

[24] NUSRAT, Ismoilov; JANG, Sung-Bong. A comparison of regularization techniques in deep neural networks. Symmetry, v. 10, n. 11, p. 648, 2018.

[25] PEDRESCHI, F.; MERY, D.; MARIQUE, T. Quality evaluation and control of potato chips. In: **Computer vision technology for food quality evaluation**. Academic Press, 2016. p. 591-613.

[26] PRANTO, Md; FARUK, Omar. **Object detection and tracking.** arXiv preprint arXiv:2502.10310, 2025.

[27] SAADI, Zena M. et al. Enhancing Convolutional Neural Network for Image Retrieval. **Journal of Intelligent Systems & Internet of Things**, v. 14, n. 2, 2025.

[28] SHIN, Hoo-Chang et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging, v. 35, n. 5, p. 1285-1298, 2016.

[29] SMITH, Alvy Ray. Color gamut transform pairs. **ACM Siggraph Computer Graphics**, v. 12, n. 3, p. 12-19, 1978.

[30] SOLEM, J. E. **Programming Computer Vision with Python**: Tools and Algorithms for Analyzing Images. Sebastopol, CA: O'Reilly Media, 2012.

[31] TASPINAR, Gulsen; KOKLU, Murat. Detecting Industrial Potato Chips Defects with Machine Learning Methods Using Deep Features. **Intelligent Methods In Engineering Sciences**, v. 3, n. 2, p. 63-69, 2024.

[32] TERRAS, Nuno et al. Integration of Deep Learning Vision Systems in Collaborative Robotics for Real-Time Applications. **Applied Sciences (2076-3417)**, v. 15, n. 3, 2025.

[33] SZELISKI, Richard. **Computer vision: algorithms and applications**. Springer Nature, 2022.

[34] VAN ROSSUM, Guido; DRAKE, Fred L. **Introduction to python 3: python documentation manual part 1**. CreateSpace, 2009.

[35] VAUGHAN, Lee. Python Tools for Scientists: An Introduction to Using Anaconda, JupyterLab, and Python's Scientific Libraries. No Starch Press, 2023.

CAPÍTULO

09 Inovações tecnológicas para automação avançada

Diego Câmara Sales[1]

9.1. CARACTERIZAÇÃO DA DISCIPLINA

Esta disciplina introduz os conceitos e métodos para formar profissionais capazes de impulsionar inovação tecnológica em regiões que possuem limitações de recursos, e vai muito além de transmitir conhecimento técnico para criar os alicerces fundamentados na teórica e prática.

Na região amazônica, onde desafios logísticos, infraestruturais e ambientais impõem limitações que exigem a aplicação de abordagens educacionais inovadoras que alinham aos anseios de um mercado competitivo e mutável, preparando os futuros engenheiros e empreendedores no domínio de temas atuais em busca de estimular o ecossistema da inovação tecnológica do estado do Amazonas.

Com isso, neste capítulo é apresentado o framework denominado IP-PBL (*Intellectual Property-enhanced Project Based Learning*). Tal abordagem se fundamenta na Aprendizagem Baseada em Projetos (PBL - *Project-Based Learning*) e conceitos de propriedade intelectual com foco em projetos inovadores de base tecnológica aplicados a desafios da região amazônica.

9.1.1. EMENTA

A ementa fundamentou-se na premissa de que aplicação efetiva do framework demanda consolidação prévia de bases de conhecimento técnico específicas que constituem o domínio sobre o qual a inovação será desenvolvida. O domínio escolhido foi automação avançada com foco em aplicações amazônicas, subdividido em quatro áreas tecnológicas complementares com distribuição estratégica de carga horária conforme apresentado na Tabela 1.

[1] Doutor em Engenharia de Automação e Sistemas. Docente da Disciplina Inovações Tecnológicas para Automação Avançada. Professor do Instituto Federal de Educação, Ciência e Tecnologia – IFAM. E-mail: diego@ifam.edu.br

Tabela 1 - Ementa do Módulo e Distribuição de Carga Horária

Tópico	Conteúdo	Carga Horária	Percentual
0	Inovação, Propriedade Intelectual e Boas Práticas em Projetos Tecnológicos	Transversal	-
1	Robótica e Inteligência Artificial em Ambientes Industriais	4 horas	20%
2	Integração IoT e Modelos Digitais Virtuais	4 horas	20%
3	Soluções Sustentáveis para Manufatura e Eficiência Energética	4 horas	20%
4	Normas e Padrões de Automação Aplicados à Realidade Amazônica	8 horas	40%
Total		**20 horas**	**100%**

O Tópico 0 foi abordado transversalmente, integrando busca de patentes, análise de anterioridade e gestão de projetos inovadores às atividades de cada domínio tecnológico.

A alocação de quarenta por cento da carga horária ao Tópico 4 reflete a compreensão de que conformidade normativa constitui habilitador crítico para desenvolvimento de inovações com elevado nível de maturidade tecnológica. As normas abordadas incluíram ISA-101 para interfaces homem-máquina, ISA-106 para automação de procedimentos, ISA-18.2 para gerenciamento de alarmes, IEC 61131 para programação de controladores lógicos e NR-12 para segurança de máquinas. Esta ênfase diferencia o módulo de abordagens tradicionais que tratam normas como conteúdo acessório, posicionando-as como elemento central para viabilização comercial de inovações.

9.1.2. OBJETIVOS

Capacitar os participantes no entendimento dos principais conceitos e na utilização de técnicas avançadas de automação e inovação tecnológica, desenvolvendo habilidades práticas para implementar e gerenciar soluções voltadas à automação industrial e eficiência energética.

Mais especificamente:

- Apresentar tecnologias emergentes, como robótica, inteligência artificial e sistemas integrados de automação, destacando suas aplicações práticas.
- Desenvolver habilidades para identificar oportunidades de aplicação dessas tecnologias no contexto da indústria local.
- Capacitar para adaptação e implementação de soluções tecnológicas sustentáveis que otimizem processos e garantam a preservação do meio ambiente.
- Fomentar a elaboração de projetos de automação industrial aplicados à realidade amazônica, integrando conceitos de eficiência energética e sustentabilidade.

9.1.3. CARGA HORÁRIA:

Carga Horária Teórica: 12h

Carga Horária Prática: 8h

Total: 20h

9.2. CONTEÚDOS

9.2.1. INTEGRANDO PBL E PROPRIEDADE INTELECTUAL NO CONTEXTO AMAZÔNICO

A aprendizagem baseada em projetos consolidou-se como metodologia pedagógica particularmente eficaz para formação de engenheiros e tecnólogos, promovendo desenvolvimento simultâneo de competências técnicas e transversais através da resolução de problemas complexos e autênticos.

Kolmos, Fink e Krogh (2004) demonstram que o PBL, quando adequadamente estruturado, desenvolve não apenas conhecimento técnico, mas também capacidades de trabalho colaborativo, pensamento crítico e aprendizagem autônoma. Mills e Treagust (2003) argumentam que a eficácia do PBL é maximizada quando os projetos abordam problemas reais, com relevância percebida pelos estudantes, e quando há equilíbrio adequado entre estruturação docente e autonomia discente.

No contexto amazônico, a aplicação de metodologias ativas ganha contornos específicos que amplificam tanto os desafios quanto as oportunidades pedagógicas. A região enfrenta limitações únicas que tornam a simples replicação de soluções tecnológicas convencionais inadequada ou inviável.

Distâncias continentais entre unidades produtivas e centros de suporte técnico criam necessidades específicas de autonomia operacional e manutenção remota. A infraestrutura logística, dependente de transporte fluvial sujeito a variações sazonais extremas, impõe restrições temporais e de planejamento que afetam diretamente a implementação de tecnologias avançadas. A variabilidade morfológica significativa de produtos regionais, como castanha-do-pará, açaí e pescado, dificulta a aplicação direta de sistemas de automação projetados para produtos padronizados. Adicionalmente, a escassez de mão de obra especializada em localidades remotas demanda soluções tecnológicas que sejam não apenas eficientes, mas também intuitivas e facilmente treináveis para operadores com diferentes níveis de qualificação.

Essas características específicas transformam desafios aparentemente limitadores em oportunidades únicas de inovação contextualizada. A necessidade de adaptar tecnologias para resolver problemas locais cria espaço para desenvolvimento de soluções originais com potencial de proteção intelectual e aplicação comercial.

Como argumentam Cassiolato e Lastres (2003) em sua análise de sistemas regionais de inovação, desenvolvimento tecnológico em regiões periféricas não deve buscar simplesmente replicar soluções de centros avançados, mas sim desenvolver capacidade de adaptação criativa que considere especificidades locais, transformando limitações aparentes em oportunidades de inovação diferenciada.

A integração de propriedade intelectual como elemento pedagógico central representa inovação metodológica com fundamentação teórica sólida que se alinha perfeitamente com as necessidades de formação para contextos regionais desafiadores. Grimaldi et al. (2011) e Thursby e Thursby (2011) demonstram que exposição sistemática a patentes durante formação acadêmica desenvolve competências essenciais para atuação inovadora: capacidade de identificar oportunidades tecnológicas, compreensão sobre proteção intelectual, pensamento inventivo estruturado e consciência sobre viabilidade comercial de inovações.

Aristodemou e Tietze (2018) argumentam que competência em análise de patentes constitui habilidade profissional fundamental, permitindo que engenheiros evitem reinventar soluções existentes, identifiquem tendências tecnológicas emergentes e posicionem suas contribuições no contexto do estado da arte.

A abordagem pedagógica adotada busca integrar a aprendizagem baseada em projetos com análise de propriedade intelectual através de ciclo metodológico aplicado consistentemente ao domínio, neste caso, voltado a automação industrial aplicado aos desafios na Amazônia. Cada tópico tecnológico do domínio escolhido fornece a fundamentação teórica e contextualização para prover ferramentas que auxiliam nas soluções aos desafios regionais fundamentados na busca e análise de patentes relevantes, desenvolvimento de projetos que incorporem elementos inovadores identificados através da análise de propriedade intelectual e, finalmente, avaliação crítica que considera não apenas viabilidade técnica, mas também originalidade e potencial de proteção intelectual.

Esta estrutura cíclica fundamenta-se nos trabalhos de Sternberg (2006) sobre desenvolvimento de criatividade em contextos educacionais, que argumenta que inovação pode ser sistematicamente estimulada através de ambientes pedagógicos apropriados que combinem conhecimento de domínio, habilidades de pensamento criativo e motivação intrínseca.

A capacitação em busca e análise de patentes constituiu componente essencial da elaboração de projetos de base tecnológica inovadores. Com isso, os alunos podem aprender a utilizar as bases de dados do Instituto Nacional da Propriedade Industrial (INPI) e da Organização Mundial da Propriedade Intelectual (WIPO), estratégias de busca utilizando palavras-chave, códigos da Classificação Internacional de Patentes (IPC) e combinações booleanas.

Desta forma, a busca pode ampliar a capacidade de interpretar a estrutura de documentos de patentes, particularmente as reivindicações, que definem o escopo legal de proteção. Como explica Risch (2012), reivindicações independentes estabelecem a invenção em sua forma mais ampla, enquanto reivindicações dependentes especificam características adicionais, criando camadas de proteção. A compreensão desta estrutura é fundamental para identificar oportunidades de inovação incremental, pois permite reconhecer quais características são essenciais à invenção protegida e quais são opcionais, revelando espaços para aprimoramentos que podem constituir modelos de utilidade ou novas patentes.

A Lei brasileira de Propriedade Industrial, Lei nº 9.279/96 (BRASIL, 1996), define modelo de utilidade como objeto de uso prático que apresente nova forma ou disposição, envolvendo ato inventivo, que resulte em melhoria funcional. Esta categoria de proteção representa oportunidade significativa para inovação em contextos regionais, pois permite proteger aprimoramentos funcionais de tecnologias existentes, adaptações para aplicações específicas e integrações criativas de soluções conhecidas.

Como argumentam Ernst (2003) e Breitzman e Thomas (2015), aproximadamente 80% do conhecimento técnico mundial está contido exclusivamente em documentos de patentes, nunca sendo publicado em periódicos científicos, o que evidencia a importância crítica de capacitar profissionais para acessar, interpretar e utilizar estas informações como base para desenvolvimento de inovações.

A experiência pedagógica demonstrou que alunos, quando equipados com ferramentas adequadas de busca e análise de patentes, e quando direcionados a

problemas contextuais específicos que conhecem profissionalmente, são capazes de identificar sistematicamente oportunidades de inovação que combinam viabilidade técnica, originalidade e potencial de aplicação comercial.

A integração de conhecimento técnico avançado em automação industrial com análise estratégica de propriedade intelectual e consciência profunda de desafios regionais específicos cria condições para desenvolvimento de inovações genuínas, caracterizadas não pela novidade radical de tecnologias empregadas, mas pela originalidade de sua integração e adaptação para contextos específicos.

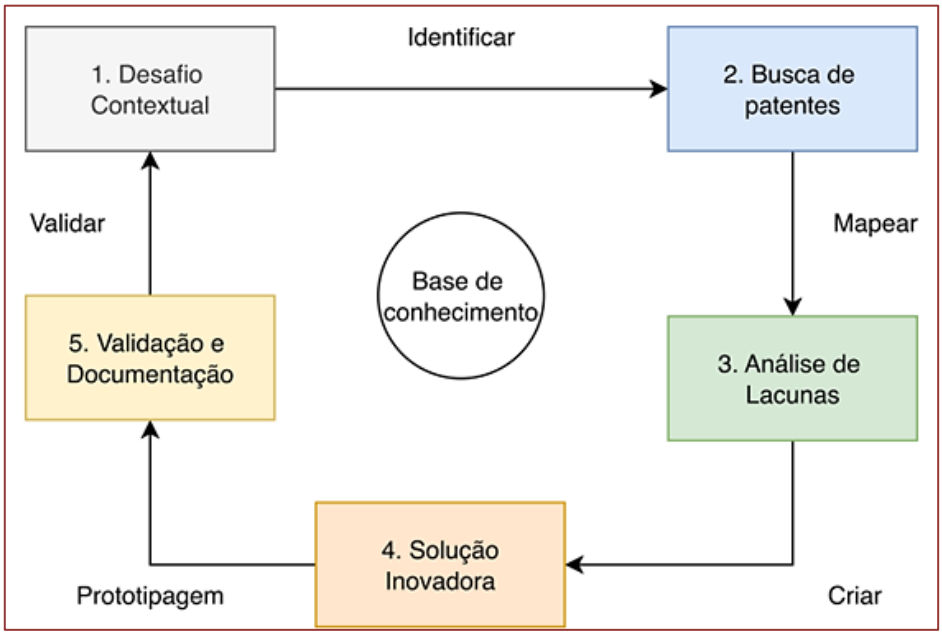
9.2.2. FRAMEWORK IP-PBL: OS 5 PASSOS DA INOVAÇÃO CONTEXTUAL

O Framework IP-PBL estrutura um processo circular e iterativo composto por cinco passos interconectados, cada um contribuindo essencialmente para o desenvolvimento de competências específicas de uma base de conhecimento que se deseja inovar, enquanto mantém coerência metodológica do conjunto conforme apresentado na Figura 9.1.

O ciclo inicia com a identificação de desafios contextuais relevantes, progride através de imersão sistemática no estado da arte via busca de patentes, identifica lacunas e oportunidades de inovação, desenvolve soluções originais e culmina em validação multidimensional que considera não apenas viabilidade técnica, mas também originalidade, conformidade normativa e potencial de aplicação comercial.

Esta estrutura cíclica permite que cada iteração refine e aprofunde compreensão dos alunos sobre o processo de inovação sistemática, transformando conhecimento técnico em capacidade efetiva de desenvolvimento de soluções originais.

Figura 9.1 – Framework IP-PBL - Ciclo de Inovação Contextual em 5 Etapas



Fonte: O autor, 2025.

O primeiro passo do framework fundamenta-se no princípio de que identificar o problema certo é mais importante que encontrar a solução perfeita para o problema errado. Esta fase, estrutura-se em três momentos sequenciais que progressivamente refinam a compreensão dos desafios relevantes.

No momento inicial de brainstorming, cada aluno, ou grupo, lista individualmente entre três e cinco problemas com foco na temática definida, neste caso automação focada em problemas Amazônicos. As ideias são apresentadas aos grupos e professor avaliador. Vale ressaltar que a base de conhecimento já deve estar consolidada para que os alunos identifiquem potenciais aplicações de tecnologias nos desafios definidos.

O segundo momento concentra-se em priorização através de ferramentas estruturadas, particularmente a Matriz de Impacto versus Viabilidade, que permite posicionar cada desafio identificado considerando seu potencial de impacto, viabilidade técnica e econômica.

O terceiro momento dedica-se ao detalhamento estruturado dos desafios selecionados através do preenchimento do Canvas de Problemas Regionais, ferramenta que orienta grupos a explicitarem sistematicamente características essenciais do problema, incluindo contexto em que ocorre, stakeholders afetados, métricas que permitiriam avaliar eficácia da solução inovadora e tecnologias a serem aplicadas.

O segundo passo os grupos realizam uma pesquisa nas bases de dados de patentes (Google Patentes, INPI e WIPO), interpretando documentos técnico-legais complexos relacionados a problemática escolhida. Os estudantes efetuam a busca através de palavra-chave e código de classificação internacional de patentes (IPC) relacionados a sua problemática (podendo ser processo ou produto). Este passo visa desenvolver a habilidade crítica para inovação sistemática através da compreensão básica de estrutura de patentes até análise de reivindicações e documentação estruturada de achados.

Como resultado, uma tabela com potenciais patentes relacionadas são listadas, identificando suas reinvindicações, tecnologias e potenciais lacunas a serem analisadas no próximo passo. Esta documentação estruturada torna-se base essencial para análise de lacunas no passo subsequente.

Após mapear o estado da arte através de análise sistemática de patentes, o terceiro passo dedica-se a identificar exatamente onde soluções existentes falham e onde residem oportunidades concretas de inovação. Esta fase, estrutura-se em três momentos progressivos que transformam conhecimento acumulado sobre patentes existentes em compreensão de oportunidades de contribuição original.

O primeiro momento requer que cada grupo apresente resumo consolidado das cinco a dez patentes analisadas, identificando padrões comuns que revelam abordagens dominantes no domínio tecnológico investigado. O Mapeamento de evolução temporal das patentes revela trajetória de desenvolvimento da tecnologia, evidenciando tendências emergentes e áreas de concentração de esforço inventivo.

O segundo momento, foca especificamente em identificação de limitações. Para cada patente analisada, os grupos devem listar explicitamente o que a solução não resolve, quais contextos não são cobertos, que restrições operacionais são mencionadas e que simplificações são assumidas. Análise detalhada de reivindicações dependentes revela quais características são opcionais na visão do inventor, sugerindo áreas onde modificações podem não infringir proteção existente.

Identificação de contextos não cobertos por soluções existentes é particularmente relevante quando se consideram aplicações regionais específicas, onde condições operacionais podem diferir substancialmente de contextos em que tecnologias foram originalmente desenvolvidas.

Com lacunas claramente identificadas, o quarto passo dedica-se ao desenvolvimento efetivo de soluções, representando o momento de maior investimento de tempo e recursos no framework. O objetivo explícito é desenvolver um esboço da solução considerando o uso das tecnologias da base de dados e que possibilite soluções que alcancem TRL 6 ou 7, caracterizando protótipos funcionais em ambiente relevante ou demonstrações de sistema em ambiente operacional, níveis de maturidade tecnológica que demonstram a viabilidade técnica.

O quinto passo é focado na validação e documentação da proposta inovadora desenvolvida. O processo de validação é focado na verificação de normas, padrões e homologações necessárias para evolução do nível de maturidade tecnológica conforme o domínio da solução. A documentação é pilar fundamental para registro da propriedade intelectual, manuais, lista de componentes, arquivos oriundos do desenvolvimento do software, hardware, automação e elétrica que auxiliam no processo de transferência de tecnologia ou futura implementação do MVP.

A aplicação do Framework IP-PBL é centrada a uma base de conhecimento/domínio/tecnologia que se deseja inovar. Como estudo de caso, o domínio escolhido foi de automação avançada com temas como:

1. Digitalização industrial – Internet das coisas, gêmeos digitais;
2. Robótica e inteligência artificial;
3. Tecnologias sustentáveis na manufatura;
4. Conformidades, normas e padrões técnicos, maturidade tecnológica e de manufatura.

9.2.3. ESTUDO DE CASO

A implementação prática do Framework IP-PBL ocorreu no módulo "Inovações Tecnológicas para Automação Avançada", componente curricular optativo do curso de especialização em Robótica Avançada oferecido pelo Instituto Federal do Amazonas.

Realizado entre 22 e 26 de setembro de 2025, o módulo integrou vinte horas de atividades intensivas estruturadas em sessões diárias de quatro horas que combinaram fundamentação teórica, análise de propriedade intelectual e desenvolvimento de projetos. A turma foi composta por dezessete alunos com perfis heterogêneos, incluindo engenheiros atuantes em indústrias do Polo Industrial de Manaus, técnicos de manutenção, estudantes do curso engenharia de automação e tecnólogo em mecatrônica industrial, proporcionando diversidade de perspectivas e experiências contextuais que enriqueceram significativamente o processo de aprendizagem.

A estruturação curricular fundamentou-se na premissa de que aplicação efetiva do framework demanda consolidação prévia de bases de conhecimento técnico específicas que constituem o domínio sobre o qual a inovação será desenvolvida. O domínio escolhido foi automação avançada com foco em aplicações amazônicas, subdividido em quatro áreas

tecnológicas complementares com distribuição estratégica de carga horária conforme apresentado na Tabela 1.

Durante o módulo, dezessete alunos organizados em seis grupos desenvolveram projetos aplicando o Framework IP-PBL aos domínios tecnológicos abordados. Três projetos destacaram-se pela originalidade das soluções, profundidade da análise de propriedade intelectual e viabilidade técnica das inovações desenvolvidas, exemplificando efetividade da metodologia proposta.

9.2.3.1. PROJETO 1: TRANSIÇÃO ENERGÉTICA PARA MATRIZ 100% RENOVÁVEL

O Grupo 4, desenvolveu proposta de transição energética para matriz 100% renovável em contexto industrial amazônico. O mapeamento de desafios contextuais identificou oportunidade relacionada ao aproveitamento passivo de luz natural e ventilação através de janelas industriais para lanternins, solução arquitetônica raramente integrada sistematicamente a estratégias amplas de automação e eficiência energética.

A busca de patentes utilizando códigos IPC F24F para climatização, E06B para janelas e H05B para iluminação revelou quarenta e três patentes potencialmente relevantes. A análise de doze patentes selecionadas demonstrou que maioria focava em otimização energética para contextos temperados onde aquecimento representa carga significativa, não considerando cenários tropicais onde ventilação natural poderia substituir climatização artificial durante maior parte do ano operacional.

Esta lacuna direcionou desenvolvimento de sistema de controle que prioriza ventilação natural através de lanternins automatizados sempre que condições de temperatura, umidade e qualidade de ar permitam, recorrendo a climatização artificial apenas quando necessário. A arquitetura proposta integrou rede de sensores IoT monitorando parâmetros ambientais, sistema de controle determinando momento ótimo para transição entre modos operacionais e atuadores para abertura automatizada de lanternins com redundância de segurança conforme NR-12.

9.2.3.2. PROJETO 2: COSMÉTICOS SUSTENTÁVEIS CERTIFICÁVEIS

O Grupo 5, desenvolveu proposta de linha de cosméticos sustentável utilizando ingredientes amazônicos com conformidade a certificações internacionais. O mapeamento identificou que ingredientes como óleos de andiroba, copaíba, buriti e manteiga de cupuaçu são subaproveitados comercialmente devido a dificuldades de processamento que atendam requisitos de certificações como Ecocert, Cosmos Organic e certificação vegana.

A análise revelou barreiras incluindo ausência de processos certificados, dificuldade em demonstrar rastreabilidade completa e variabilidade de qualidade devido a métodos artesanais. A busca de patentes utilizando códigos IPC A61K para preparações cosméticas, A61Q para uso específico e C11B para óleos identificou dezessete patentes brasileiras e cento e quarenta e três internacionais relacionadas a óleos amazônicos.

A análise demonstrou que enquanto formulações são extensivamente patenteadas, processos de manufatura automatizados garantindo rastreabilidade completa e conformidade simultânea com múltiplas certificações não são adequadamente abordados. Esta lacuna direcionou desenvolvimento de plataforma de manufatura automatizada integrando três elementos inovadores.

Primeiro, o sistema de rastreabilidade baseado em blockchain registrando cada etapa desde recebimento de matérias-primas até produto, criando registros imutáveis auditáveis que atendem requisitos de certificações. Segundo, a automação de procedimentos conforme ISA-106 garantindo que processos sigam rigorosamente métodos permitidos por certificações, modelados utilizando *Sequential Function Charts* especificando sequência de operações e parâmetros críticos. Terceiro, o controle de qualidade automatizado utilizando visão computacional para verificar cor, textura e homogeneidade, e espectroscopia no infravermelho próximo para verificação não destrutiva de concentrações de ingredientes ativos.

9.2.3.3. PROJETO 1: TRANSIÇÃO ENERGÉTICA PARA MATRIZ 100% RENOVÁVEL

O Grupo 6 desenvolveu solução para universalização de energia em comunidades isoladas com gestão integrada de ciclo de vida de equipamentos fotovoltaicos. O mapeamento identificou como caso específico a Reserva de Desenvolvimento Sustentável do Rio Negro em Manacapuru, composta por três comunidades totalizando duzentas e noventa e oito pessoas sem acesso à rede elétrica.

Os alunos documentaram características específicas incluindo construções palafíticas, ausência de infraestrutura de saneamento e gestão de resíduos, e acesso apenas por transporte fluvial sujeito a variações sazonais extremas. A busca de patentes utilizando códigos IPC H02J para fornecimento de energia, H01L para dispositivos fotovoltaicos e H01M para baterias identificou doze patentes brasileiras e oitenta e seis internacionais relacionadas a sistemas off-grid.

A análise revelou que maioria foca em aspectos técnicos de geração e armazenamento, mas não aborda gestão de resíduos eletrônicos em contextos de difícil acesso logístico. Investigação do relatório do Instituto de Energia e Meio Ambiente revelou que o programa Mais Luz para Amazônia gerará até 2050 aproximadamente quatro mil toneladas de resíduos eletrônicos demandando coleta e destinação adequada em regiões com infraestrutura extremamente limitada.

A Política Nacional de Resíduos Sólidos estabelece responsabilidade compartilhada, mas mecanismos práticos para implementação em comunidades ribeirinhas isoladas não estão adequadamente desenvolvidos. A solução proposta propôs o desenvolvimento de um barco-oficina móvel equipado com ferramentas para instalação e manutenção, espaço climatizado para equipamentos novos e área segregada para resíduos, operando em circuito regular visitando comunidades semestralmente.

Adicionalmente, o sistema realiza a rastreabilidade digital registrando cada equipamento com identificação única, localização GPS e histórico de manutenções, permitindo planejamento logístico otimizado. A busca de anterioridade específica para barco-oficina móvel não identificou patentes equivalentes integrando oficina fluvial para gestão de ciclo de vida de sistemas fotovoltaicos. Os alunos argumentaram que esta integração, considerando adaptações para navegação com variações sazonais extremas e projeto de armazenamento seguro em ambiente fluvial, poderia constituir modelo de utilidade com aplicação significativa para programas similares em outras regiões com comunidades ribeirinhas.

9.2.4. RESULTADOS E IMPLICAÇÕES

A análise comparativa dos três projetos revela padrões consistentes validando eficácia do Framework IP-PBL. Os três demonstraram capacidade de identificar desafios contextuais específicos através de mapeamento sistemático e contextualização de desafios no domínio proposto.

A busca de patentes foi conduzida com o professor capacitado que, refinando através de códigos IPC específicos e palavras chaves, pode analisar com os alunos as lacunas de automação em processos e produtos amazônicos. A análise crítica permitiu compreensão do estado da arte e identificação de oportunidades de contribuição original, demonstrando que alunos desenvolveram capacidade de posicionar propostas no contexto do conhecimento existente. As soluções integraram conhecimentos de múltiplos tópicos do módulo, demonstrando síntese e aplicação integrada.

A experiência revelou particularidades importantes sobre processo de inovação em contextos regionais. Diferentemente de inovação em centros avançados onde novidade reside em tecnologias fundamentalmente novas, inovação contextual caracteriza-se pela originalidade na integração e adaptação de tecnologias estabelecidas para resolver problemas específicos.

Esta forma é particularmente relevante para regiões periféricas como Amazônia, onde desafios únicos criam oportunidades para soluções diferenciadas que, embora utilizem componentes conhecidos, constituem integrações originais com potencial de proteção através de modelos de utilidade. Como argumentam Ernst (2003) e Breitzman e Thomas (2015), aproximadamente 80% do conhecimento técnico mundial está contido exclusivamente em documentos de patentes, nunca sendo publicado em periódicos científicos, o que evidencia a importância crítica de capacitar profissionais para acessar, interpretar e utilizar estas informações como base para desenvolvimento de inovações

A aplicação do framework demonstra que integração sistemática de aprendizagem baseada em projetos com análise estratégica de propriedade intelectual constitui abordagem pedagógica eficaz para formação de profissionais simultaneamente competentes tecnicamente, conscientes de desafios contextuais e capazes de inovar sistematicamente.

Os três projetos exemplificam como esta metodologia transforma alunos de receptores passivos em investigadores ativos do estado da arte, identificadores de lacunas tecnológicas e proponentes de soluções fundamentadas em análise técnica inovativa.

Para região amazônica, capacitar profissionais para identificar oportunidades através de adaptação criativa de tecnologias globais para contextos locais representa estratégia essencial para construção de sistemas regionais de inovação capazes de gerar soluções adaptadas com potencial de proteção intelectual e aplicação comercial sustentável.

REFERÊNCIAS

[1] ARISTODEMOU, L.; TIETZE, F. The state-of-the-art on Intellectual Property Analytics (IPA): A literature review on artificial intelligence, machine learning and deep learning methods for analysing intellectual property (IP) data. **World Patent Information**, v. 55, p. 37-51, 2018.

[2] BRASIL. Lei nº 9.279, de 14 de maio de 1996. Regula direitos e obrigações relativos à propriedade industrial. **Diário Oficial da União**, Brasília, DF, 15 maio 1996.

[3] BREITZMAN, A.; THOMAS, P. The emerging clusters model: A tool for identifying emerging technologies across multiple patent systems. **Research Policy**, v. 44, n. 1, p. 195-205, 2015.

[4] CASSIOLATO, J. E.; LASTRES, H. M. M. O foco em arranjos produtivos e inovativos locais de micro e pequenas empresas. In: LASTRES, H. M. M. et al. (Org.). Pequena empresa: cooperação e desenvolvimento local. Rio de Janeiro: **Relume Dumará**, 2003. p. 21-34.

[5] ERNST, H. Patent information for strategic technology management. **World Patent Information**, v. 25, n. 3, p. 233-242, 2003.

[6] GRIMALDI, R. et al. 30 years after Bayh-Dole: Reassessing academic entrepreneurship. **Research Policy**, v. 40, n. 8, p. 1045-1057, 2011.

[7] KOLMOS, A.; FINK, F. K.; KROGH, L. (Eds.). The Aalborg PBL Model: Progress, **Diversity and Challenges**. Aalborg: Aalborg University Press, 2004.

[8] MILLS, J. E.; TREAGUST, D. F. Engineering education: Is problem-based or project-based learning the answer? **Australasian Journal of Engineering Education**, v. 3, n. 2, p. 2-16, 2003.

[9] RISCH, M. The Shape of Things to Come: What We Can Learn from Patent Claim Length. **Santa Clara Computer & High Tech. Law Journal**, v. 28, n. 2, 2012.

[10] STERNBERG, R. J. The nature of creativity. **Creativity Research Journal**, v. 18, n. 1, p. 87-98, 2006.

[11] THURSBY, J. G.; THURSBY, M. C. Faculty participation in licensing: Implications for research. **Research Policy**, v. 40, n. 1, p. 20-29, 2011.

www.poisson.com.br
contato@poisson.com.br

@editorapoisson
https://www.facebook.com/editorapoisson